1991年２月１日発行（毎月１回１日発行）第10巻２号通巻106号　昭和58年11月２日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集1 グラフィックの"実験的"手法

特集2 SX-WINDOWプログラミング
1990年度GAME OF THE YEARノミネート発表
カードゲーム スロットポーカー

2
1991

SOFT BANK オー！エックス
定価560円

SHARP

X68000 SUPER登場

このたび新たにラインアップされた“X68000 SUPER”は、すでに発売されている“SUPER HD”と同様、SCSIインターフェイスを標準装備しています。また、その他のシリーズにはオプションとしてSCSIボード(CZ-6BS1)がサポートされ、大容量外部記憶装置をはじめ、各種SCSI装置との接続が可能になったのは、ご存じのとおりです。

SCSI規格とは……

SCSIは1986年にANSI(米国規格協会)で規格化された仕様で、Small Computer System Interfaceの略。小型コンピュータの周辺機器接続のための世界共通の規格です。大容量外部記憶装置(大容量ハードディスク、CD-ROM、DATなど)に加え、登場が期待される高速スキャナ、次世代プリンタなどのSCSI装置を、デイジーチェーン方式で最大7台まで接続可能。大容量データの高速転送、および単一のインターフェイスでの周辺機器の複数制御が特長です。

X68000と大容量メディア

サウンドクリエーション&コンピュータグラフィックス。X68000のオハコともいうべきこの領域は、感性あふれるユーザーにとって最も魅力的である反面、表現の繊細さに比例して必要な外部記憶容量も増大します。サンプリング、MIDI、レイトレ……。その潜在能力をフルに引き出すには、大容量メディアへの対応が必須です。たとえば、新発売の光磁気ディスク(CZ-6MO1)と光磁気ディスクカートリッジ(JY-701MPA)なら、ディスク1枚で65,536色画像にして1,000枚強、15.6kHzの音声サンプリングデータで約20時間強もの情報を記憶できます。絵に書いた餅とされていた「画像データベース」も、「サンプリングシステム」さえも、もう実用レベル。SCSIの採用が、夢の大容量メディアに応えてくれるからです。

▲MOディスクドライブ情報

▲MOドライブディレクトリ情報

▲MOディスクのフォーマット

●デイジーチェーン方式による接続例

SCSIボードCZ-6BS1▶
標準価格29,800円(税別)

X68000の先見性

初代X68000は、すでにハードディスクインターフェイスを内蔵していたこと。当時まだ一般的ではなかったハードディスクに対して先見の発想で臨んでいたわけです。今回のSCSI対応も同様、100MBを超える大容量メディアハンドリングがスタンダードになる日も、そう遠くはありません。

大容量ハードディスクか？ 光磁気ディスクか？ それとも……

考えもしなかった新しいデバイスか。新製品X68000SUPERのSCSIインターフェイスに何を接続するかは、賢明なユーザー諸兄にお任せするとして…。このマシンがまた新たな一歩を踏み出したことに異論はないはずです。蛇足ながらこのSUPERシリーズに関していわせてもらえれば、その日から大容量ハンドリングをお望みの方にはSUPER HDを。未来に夢を託したユーザーはSUPER、といったところでしょうか。

＊SCSI装置をご使用の場合は、Human68k Ver2.0以上でご使用ください。
＊ビジュアルシェル上からはSCSI装置はご使用になれません。

シャープX68000パソコン教室開催中
- 会場：市ヶ谷教室 シャープ東京支社ビル
- コース：入門コース・表集計コース・音楽コース・絵画コース
- 申込受付電話番号(03)3260-8365
- 受講料：2,000円(税別)

資料請求券 X68000 Oh!X 2体

本体同梱の入会申込ハガキを送るだけで、無料入会。
メリット1：会員No.入り、オリジナル**会員証電卓**がもらえる。
メリット2：各種フェアご優待・イベントご案内等、数々の特典アリ。
メリット3：10月1日スタート！ X68000の活用情報が手に入る「EXEおみこし活動」に参加できる!!

ステップアップサービス(有料)
「おみこしかつぎ人」制度も新設
EXEおみこし活動のお問い合わせは、X68000EXEクラブ「おみこし活動隊」まで
☎(06)886-0354
詳細はX68000販売店店頭で
ポスター・おみこしPressをご覧下さい。

敢えてX68000の大容量メディア対応を実証する意味。

カラー画像ファイル、サンプリングファイルへ。
X68000のクリエイティブユースに待望の大容量ファイル。
書き換え可能な光磁気ディスク、登場。

■光磁気ディスクユニット■
CZ-6MO1…標準価格450,000円(税別)

■光磁気ディスクカートリッジ■
JY-701MPA…標準価格30,000円(税別)

写真のX68000とディスプレイは別売です。

SX-WINDOW、SCSIインターフェイス標準装備。

PERSONAL WORKSTATION
SUPER

PERSONAL WORKSTATION

SUPER
本体＋キーボード＋マウス・トラックボール
CZ-604C-TN(チタンブラック) 標準価格348,000円(税別) NEW
HDタイプ CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

PRO II
本体＋キーボード＋マウス
CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)
HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

- 15型カラーディスプレイテレビ(ドットピッチ0.39㎜) CZ-602D-BK(ブラック)・-GY(グレー)※ 標準価格99,800円(チルトスタンド同梱・税別)
- 15型カラーディスプレイテレビ(ドットピッチ0.39㎜) CZ-605D-BK(ブラック)・-GY(グレー) 標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
- 15型カラーディスプレイテレビ(ドットピッチ0.31㎜) CZ-613D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー) 標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
- 14型カラーディスプレイ(ドットピッチ0.31mm) CZ-603D-BK(ブラック)・-GY(グレー) 標準価格84,800円(チルトスタンド同梱・税別)
- 14型カラーディスプレイ(ドットピッチ0.31mm) CZ-604D-BK(ブラック)・-GY(グレー) 標準価格94,800円(スピーカー2個/チルトスタンド同梱・税別)
- 14型カラーディスプレイ(ドットピッチ0.31mm) CZ-606D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー) 標準価格79,800円(チルトスタンド同梱・税別) NEW
- 21型カラーディスプレイ(ドットピッチ0.52mm) CU-21HD-BK(ブラック) 標準価格148,000円(スピーカー2個同梱・税別)

※印の商品は在庫僅少です。

X68000
自分流カード
デザインコンペ
作品大募集

〈応募要領〉●応募方法/X68000で作成したポストカードサイズのデザインカードを送って下さい。(ソフトは自由)●作品分類/部門A：クリスマスカード、ニューイヤーカード 部門B：バレンタインカード、バースディカード 部門C：暑中見舞カード、サークル・趣味の会お知らせカード●賞/A・B・C各部門毎に優秀作品を選考、オリジナルカレンダーに掲載してプレゼントします。※優秀作品賞：掲載作品応募者に、カレンダー及びオリジナル表彰楯を進呈。※参加賞：応募者全員に、カレンダーを進呈。(応募作品に関わる諸権利は主催者に帰属するものとして作品の返却はいたしません)
●応募期間/1990年10月1日～1991年2月28日(消印有効)

詳細はX68000販売店店頭で、チラシ・ポスターをご覧下さい。

●お問い合わせは…
シャープ株式会社
電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部
〒162 東京都新宿区市谷八幡町8番地☎(03)3260-1161(大代表)

特集1　グラフィックの"実験的"手法

スロットポーカー

THE USER'S WORKS

プール・オブ・レイディアンス

エメラルドドラゴン

Fine Scanner-X68

Oh!X

CONT

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／岡崎栄子 浅井研二 ●協力／有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／グループごじら

表紙絵：須藤　牧人

1991 FEB. 2

E N T S

●THE SOFTOUCH

●読みもの

●連載/紹介/講座/プログラム

UNIXはAT&T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはDIGITAL RESEARCH
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS CはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
WordStar, WordMasterはWORDSTAR International
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名，CPUは一般に各メーカーの登録商標です。本文中では"TM"，"R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

SHARP

システムパフォーマンスを実証する多彩なペリフェラル。

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

カラーディスプレイ

NEW

14型カラーディスプレイ
CZ-606D-TN•-BK•-GY
標準価格79,800円(税別)
(チルトスタンド同梱)

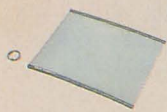
NEW

15型カラーディスプレイテレビ
CZ-605D-BK•-GY
標準価格115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

14型カラーディスプレイ
CZ-604D-BK•-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

NEW

15型カラーディスプレイテレビ
CZ-613D-TN•-BK•-GY
標準価格135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格148,000円(税別)
(スピーカー2個同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)

チューナー

RGBシステムチューナー
CZ-6TU-BK•-GY
標準価格 33,100円(税別)
(リモコン付)

アートツール

画像入力

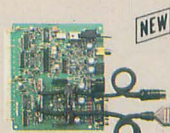
カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円(税別)

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

映像出力

NEW

ビデオボード※3
CZ-6BV1
標準価格 21,000円(税別)

プリンタ

熱転写カラープリンタ

48ドット
熱転写カラー漢字プリンタ
★CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)

NEW

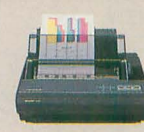
48ドット
熱転写カラー漢字プリンタ
CZ-8PC5-BK
2月発売予定

カラービデオプリンタ

カラービデオプリンタ
★CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット

カラーイメージジェット※4
IO-735X
標準価格248,000円(税別)
(信号ケーブル別売)

カラードットプリンタ

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)

24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)

ドットプリンタ

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

光磁気ディスク

光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格 450,000円(税別)
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク

ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)

増設用ハードディスクドライブ (40MB)
(CZ-602C/603C/652C/653C内蔵用)
CZ-64H※
標準価格 120,000円(税別)
(取付費別)

増設用ハードディスクドライブ (80MB)(CZ-604C内蔵
CZ-68H※
標準価格 160,000円(税別)
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D 604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。 ※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。
※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。 ※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cに使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。(但し、CZ-623Cは不要
また、X68000用OS Human68K ver.2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格

X1・X1turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21HD	148,000円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円

●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	★CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー(2本1組)標準装備、ミュージックツール同梱

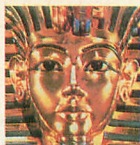
プリンタ		
●24ピンカラー漢字プリンタ(80桁)	CZ-8PG1	130,000円
●24ピンカラー漢字プリンタ(136桁)	CZ-8PG2	160,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK10	97,800円
●48ドット熱転写カラー漢字プリンタ	★CZ-8PC4	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4-GY	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC5-BK	2月発売予定
●カラービデオプリンタ	★CZ-6PV1	198,000円
●カラーイメージジェット	IO-735X	248,000円

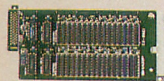
ファイル		
●ミニフロッピーディスクユニット(2HD・2D)※3	★CZ-520F	118,000円

シャープ株式会社

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代

資料請求券
X68000ペリフェラル
Oh!X
2/8

お望みのパワーシステムへ。

ボード

拡張メモリ

1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)

1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)

2MB増設RAMボード ※6
CZ-6BE2
標準価格 79,800円(税別)

4MB増設RAMボード ※6
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

NEW
SCSIボード ※7
CZ-6BS1
標準価格 29,800円(税別)
(ソフトウェア(SCSIユーティリティ)同梱)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX

FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

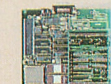
モデム

モデムユニット ※8
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

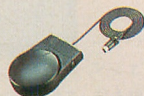
RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

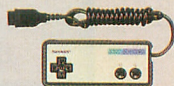
LANボード

LANボード
CZ-6BL1
標準価格268,000円(税別)
(イーサネット用)

NEW
CZ-6BL2
標準価格298,000円(税別)
(イーサネット/チーパネット両用)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力

インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)

マウス・トラックボール
CZ-8NM3
標準価格9,800円(税別)

トラックボール
CZ-8NT1
標準価格 13,800円(税別)

マウス
CZ-8NM2A
標準価格 6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット

拡張 I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック

システムラック
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。 ※7 CZ-600C、601C、602C、603C、611C、612C、613Cに装着の場合、I/Oスロット2に装着ください。CZ-652C、653C、662C、663Cに装着の場合はI/Oスロット4に装着ください。また、CZ-6BG1、6BU1、6BL1、6BL2、6BN1などのボードは、接続コネクタとの関係で本ボードとの併用はできませんのでご注意ください。なお、本ボードはX68000用OS Human 68K ver.2.0以上にてご使用ください。 ※8 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D) ※4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード ※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス ※6	CZ-8BF1	14,800円
●JIS第1水準漢字ROM ※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張 I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード ※10	CZ-8BN1	27,800円

●品番中の－表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 14/15型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地☎(03)3260-1161(大代表)

SHARP

ハイアビリティを実証する多彩なソフトウェア。

ドロー編集、WYSIWYG印刷、こんなC.G.ツールが欲しかった。

本格的なロゴタイプやPOPを簡単に作成できるグラフィックツールです。優先順位が任意に指定できるドローセル、ペイントセル、テキストセルの3つの仮想セルで、目的にあった自由なグラフィックが駆使できます。また印刷は、画面イメージがそのまま印刷イメージとなるWYSIWYG(What You See Is What You Get)を実現。A6/A5/A4/A3/B6/B5/B4/葉書サイズで8色カラー印字できます。

〈ドローセル〉ベジェ曲線によって少ないデータ量でも複雑な絵を描くことができます。エンベロープ変形を始めとした豊富な編集機能を持っており、拡大、縮小しても絵の美しさは変わりません。またテキストセルで作成したベクトルフォントデータを自由に変形し、オリジナルロゴタイプやPOPを作成できます。

〈ペイントセル〉ペンやエアーブラシ、ペンキなどを使って、ピクセルで構成されたビットマップ図形を描くことができます。また、「NEW PrintShop PRO-68K」や「X-BASIC」、「Z's STAFF PRO-68K」のデータ取り込みやイメージスキャナによる取り込みをサポートしています。

〈テキストセル〉通常の文字入力機能に加え、ベースライン変形などの多彩な編集機能によって自由に文字の加工ができます。また英数文字のベクトルフォントを標準装備。さらに「Z's STAFF PRO-68K Ver2.0」、「書体倶楽部」の日本語ベクトルフォントが利用可能。また、内蔵の漢字ROMフォントも自動的にベクトルフォントデータに変換しますので、簡単に日本語ロゴタイプを作成することができます。

※「Z's STAFF PRO-68K」、「書体倶楽部」は、㈱Zeitの製品です。
※本ソフトの動作には、メインメモリ2MBが必要です。

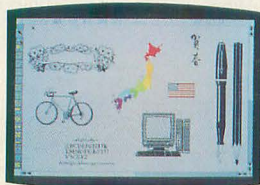
CANVAS PRO-68K CZ-249GS 標準価格29,800円(税別)

●主として個人用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

海のデータ/動物のデータ/スポーツのデータ/鳥のデータ/人物のデータ/食物のデータ/昆虫のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.1
CZ-255GS 標準価格8,800円(税別)

●主としてビジネス用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

OA関係のデータ/飾りのデータ/コンピュータ関係のデータ/POPのデータ/国旗のデータ/字体のデータ/地図のデータ/乗り物のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.2
CZ-256GS 標準価格8,800円(税別)

バージョンアップされたCコンパイラと、強力なBASTOCチェッカー。

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできる「ソースコードデバッガ」を搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(830種以上)となっています。C言語の標準であるANSI規格準拠をさらに強化。「プログラム保守ユーティリティ(MAKE)」や「ライブラリアン」など各種ツールを追加しました。その他「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「コンバータ」、などのツールが装備されています。

※C compiler PRO-68K(CZ-211LS)を既にお持ちの方は、登録カードをもとに有償バージョンアップを行います。
※本ソフトの動作にはメインメモリ2MBが必要です。

CZ-245LS 標準価格44,800円(税別)

トラブルエラーの悩み解消！「XBAStoC」の強力ツールの登場です。

X-BASICプログラムのコンパイル時、発見しづらいトラブルエラーに悩まされていたプログラムの問題点をひとつひとつ指摘。エラーとなる直接原因だけでなく、注意項目も指摘します。これにより、X-BASICでは実行できたのにコンパイルするとエラーが発生する、といったプログラムの修正が簡単にできます。

●指摘したトラブルの結果を、画面やプリンタなどの外部デバイスに簡単に出力できます。●エラーラインとエラーレポート、2つのエラーファイルを自動的に生成。●グラフィカルな画面による簡単操作。●コマンドラインからダイレクトに操作を指定。バッチファイルに組み込むなどの修正作業の自動化が可能。●GP-IBボード(CZ-6BG1)とユニバーサルI/Oボード(CZ-6BU1)付属の拡張外部関数もコンパイル可能。

※X-BASICプログラムをコンパイルするためには、別売の「C compiler PRO-68K」(CZ-211LS)または「C compiler PRO-68K ver2.0」(CZ-245LS)が必要です。

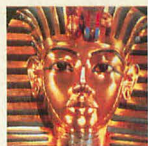
XBAStoC CHECKER PRO-68K
CZ-260LS 標準価格9,800円(税別)

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。

資料請求券 Cソフト Oh!X 2体

シャープオリジナルソフトウェア X68000

お望みのワークベンチへ。

ビジネスツール

Hyperword
■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。

CYBERNOTE PRO-68K
■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-300Lが必要)。

Stationery PRO-68K
■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-300Lが必要)。

TOP給与計算エキスパート
■CZ-228BS標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計
■CZ-227BS標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。

グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

通信ツール

Communication PRO-68K ver2.0
■CZ-257CS 標準価格19,800円(税別)

Communication PRO-68Kのバージョンアップ版です。MNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポートしています。

※バージョンアップ対応中。

開発ツール

SX-WINDOW ver1.0
■CZ-259SS 標準価格6,800円(税別)

複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。IOCSコールを利用したソフトの処理速度を高速化するIOCS.Xを付属。

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

マルチタスク機能、リアルタイム機能を活かした使いやすく機能的なOS環境を提供します。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

システムパフォーマンスをさらに高める処理機能を付加したHuman68kの最新バージョンです。

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS標準価格188,000円(税別)

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS 標準価格28,800円(税別)

MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI. 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI. 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

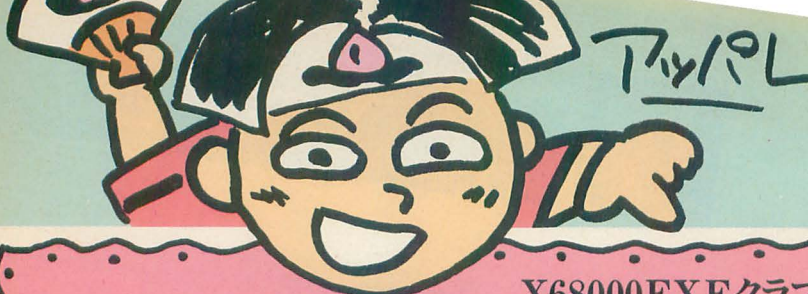
スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987

ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987

アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

アクションゲーム
〈サイバリオン〉
■CZ-229AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部サッカー編〉
■CZ-262AS
標準価格8,800円(税別)
©TECHNOS JAPAN CORP. 1990

X68000EXEクラブ

EXEおみこし活動スタート!

◉まずはEXEクラブへ◉

入会無料で3つのメリット!手続きは本体同梱の入会申込ハガキを送るだけ。

メリット1▶会員番号入りオリジナル会員証電卓がもらえます。

メリット2▶各種フェアご優待・イベント案内等、数々の特典があります。

メリット3▶X68000の活用情報が手に入る「EXEおみこし活動」に参加できます。

※「入会申込ハガキをなくしてしまった」という方は、おみこし活動隊までお電話下さい。

◉EXEおみこし活動とは?◉

いわば「X68000ユーザーの、X68000ユーザによる、X68000ユーザーのための」活動です。おみこしPRESSを通じて会員同士情報を交換し、もっと68を使いこなして盛り上がってしまおう/というワケ(モデムがなくてもできるパソコン通信のようなもの?!)なので、X68000へのラブコール、会員独自のテクニック、活用法(マニアックなものでなくても他の会員には貴重/)等あなたの68自慢をドシドシ聞かせてください。会員からのメッセージは「おみこし活動隊」が整理してコミュニケーションペーパー「おみこしPRESS」にバッチリ掲載します。

10月1日開設 **おみこし活動隊へのアクセス方法**

投稿受付/大阪市淀川区西中島1丁目9-16 新大阪ストロングビル2F
X68000EXEクラブ「おみこし活動隊」係

電話受付 **06-886-0354** FAX受付 **06-304-1539**

★受付時間……平日/昼2時〜夜8時
日祝/昼12時〜夜8時

◉そしてEXE会員究極の◉
「おみこしかつぎ人」を大募集!

「かつぎ人」とは、より積極的におみこし活動に参加する人のこと。EXE会員は「かつぎ人」になることで、X68000ユーザーとしてますます充実、3つのメリットで強力にサポートされます。

《メリット1》「おみこしかつぎ人の集い」に参加できます。
シャープとEXE会員の双方向コミュニケーションの場として開設されるX68000情報交換会「おみこしかつぎ人の集い」は、シャープの68スタッフと直に意見交換ができるおいしいチャンス。91年2月より全国レベルでスタートするこの会に参加すれば、68ユーザーとしてトップレベルです。

《メリット2》「おみこしPRESS」定期送付。
お店まで足を運ばなくても「おみこしPRESS」が毎号お手元に届きます。

《メリット3》「ソフトウェア・フィールド」直送。
X68000最新ソフト・各種周辺機器が一覧できる「ソフトウェア・フィールド」を半年に1回お送りします。

おみこしかつぎ人になるには……

以下の年会費(おみこしかつぎ代)が必要です。〈個人入会〉3,000円/〈グループ入会(5人1組)〉2,500円/1人 おみこし活動スタート記念特別割引期間(91年1月末まで)に限り〈個人入会〉2,000円 〈グループ入会(5人1組)〉1,500円/1人。

★詳細は「おみこし活動隊」にお電話ください。

カードデザインコミュニケーション

カードデザインコンペ作品大募集

X68000で、あなただけのとびっきりのデザインカードを…。「よし、これだ!」と思ったら、あとは応募するしかない!! ソフトは自由。点数無制限。アイデアを生かした楽しいカードデザインを募集します。A・B・C各部門毎に優秀作品を選考、オリジナルカレンダーに掲載してプレゼントします。多数の力作をお待ちしています!

《応募要領》

◎作品分類…部門A/クリスマスカード、ニューイヤーカード 部門B/バレンタインカード、バースディカード 部門C/暑中見舞カード、サークル・趣味の会お知らせカード。◎応募資格…X68000で制作されたカードに限ります。◎応募方法…ポストカードサイズ(10×15cm)の作品裏面に必要事項(作品名・部門名・名前・住所・X68000使用機種名・周辺機器名・ソフト名)を明記し、封筒に入れて郵便にてお送りください。◎受付期間…90年10月1日〜91年2月28日(消印有効)。◎賞…優秀作品賞/優秀作品掲載のオリジナルカレンダー及びオリジナル表彰楯を進呈。参加賞/応募者全員に優秀作品掲載のオリジナルカレンダーを進呈。※応募作品にかかわる諸権利は主催者に帰属するものとして作品の返却は致しません。◎発表…オリジナルカレンダー・オリジナル表彰楯の発送をもってかえさせていただきます。

X68000パソコンスクール開校

●会場及び申込受付問い合わせ先
シャープ大阪OAショールーム
大阪市中央区今橋3-1-7日本生命今橋ビル TEL (06)222-7655
●定員/20名(先着順)
●1月19日(土)…表集計(BUSINESS PRO-68K)の多彩な活用方法
●2月16日(土)…グラフィックツール(Z's STAFF PRO-68K Ver.2.0)
●開催時間/1日2回
13:00〜15:00、15:30〜17:30
●受講料/無料

※月1回第3火曜
X68000パソコン
初心者コース

シャープ株式会社 ●お問い合わせは…シャープ(株)電子機器事業本部システム機器営業部

C-TRACE CG コンペティション'91 開催！

◎応募要項◎

応募期間 平成3年1/20〜3/20

発　表 平成3年5月発売のASCII・Oh!Xの当社広告紙面

募集規定 C-TRACEユーザーがC-TRACEを使用して作成したCG静止画像、アニメーション

募集部門 ●静止画キャラクター部門●静止画アート部門●静止画産業デザイン部門●アニメーション部門

応募方法 キャストまで応募要項を請求後、静止画CG画像はフロッピーディスク、アニメーションはビデオテープ（VHS、ベータ、8㎜のいずれか）で応募

賞 ■グランプリ、準グランプリ、金賞、銀賞、銅賞（各1名）、■ステゴちゃん賞（3名）、■メーカー特別賞

賞　品 ■グランプリ、準グランプリ、金賞、銀賞、銅賞の順に以下の賞品の中から1つを選択――海外旅行クーポン、トランスピュータ、フレームバッファ、キャスト商品券（10万円券、7万円券、3万円券）■ステゴちゃん賞――ステゴちゃんぬいぐるみ■メーカー特別賞――SONY賞〈データディスクマン〉、アイ・オー・データ機器賞〈EMSメモリボード(2M)〉、SHARP賞〈コプロセッサボード〉、ASCII賞〈月刊アスキー1年間無料購読〉

※応募者全員にもれなくキャストオリジナルポストカードをプレゼント

審査員（敬称略） ASCII・Oh!X・Oh!PC：各編集部、CGキッチンまざあぐうす代表長谷川一光、C-TRACEユーザークラブ会長小石光、東京工学院芸術専門学校講師塩沢左千子、超人（超能力者・平成2年11/27フジTV出演）玉手峰人、イラストレーター伊川英雄、各協賛会社

協　賛 ソニーコンピュータシステム株式会社、株式会社アイ・オー・データ機器、シャープ株式会社、月刊アスキー編集部

メモリー解放宣言。
C-TRACE98 EXTENDER
価格￥128,000（税別）

●PC-9800シリーズ、PC-286, 386シリーズ●メインメモリとして最大16M使用可能●EMSによるメモリ拡張のようにスピードを犠牲にしません●30%の高速化（当社C-TRACE Ver.3.0比）●Ver.3.0からのステップアップ受付中〈ソフトウエア〉

1670万色表示。
フルカラーフレームバッファ★
価格￥69,800（税込）

●PC-9800シリーズ、PC-286, 386シリーズ●1670万色同時表示●フレームバッファ制御のためのサンプルソース付き●RAMディスクドライバ付き●ズーム、スクロール＆パン機能をハードウェアでサポート●フレームバッファ＋ユーティリティディスク（2枚）

超高速。
C-TRACE TP Ver.3.0★
価格￥298,000（税込）

●PC-9800シリーズ、PC-286, 386シリーズ、X68000シリーズ●パソコンでレイトレーシングをワークステーション並みのスピードで実行可能●並列処理によりスピードアップも可能●トランスピュータボード＋C-TRACE Ver.3.0のセット

メタボール対応。
C-TRACE＋ NEW
価格￥198,000（税別）

●PC-9800シリーズ、PC-286, 386シリーズ、X68000シリーズ●C-TRACEシリーズ最上位のスペック●メタボール対応●スポット光源対応●αチャンネル対応により高度な合成が可能●スコープ機能対応により部分的に画像の再計算が可能●DOSエクステンダにも対応●差額交換受付中〈ソフトウエア〉

C-TRACE68 Ver.3.0	￥98,000（税別）
C-TRACE98 Ver.3.0	￥98,000（税別）
C-TRACE TOWNS	￥68,000（税別）
C-TRACE NEWS Ver.3.0	￥530,000（税別）

一部クレジット取扱可

★の製品は店頭販売いたしておりません。直接当社まで、お申し込みください。

※電話・FAX番号が変わりました。お間違えのないようお願い致します。

● 株式会社キャスト●お問い合せ先 〒158 東京都世田谷区等々力2-1-13 TEL.03-3705-1065 FAX.03-3705-5224

ユーザーズテレホン ☎大阪06(315)8255
平日の午後1時半から6時の間は、お問い合わせに直接お答えします。
その他の時間と土・日・祝日はまるまる24時間録音もできるテープサービスです

◆標準価格に消費税は含まれておりません。お買上げの際に別途消費税をお支払い下さい。
◆通信販売ご希望の方は、住所・氏名・電話番号・商品名・機種名・メディアを明記の上、現金書留または郵便振替(大阪8-303340)にてお申し込み下さい。
送料は無料ですが、標準価格に消費税の3%を加えた金額をお送り下さい。

Humming Bird Soft™
株式会社エム・エー・シー ハミングバードソフト
〒530 大阪市北区曽根崎2丁目2番15号

X68000シリーズに、新登場！
シュヴァルツシルト
好評発売中！

SCENARIO SIMULATION GAME

狂嵐の銀河
Schwarzschild
シュヴァルツシルト

ストーリィ性を持ったドラマティックなゲーム展開

シュヴァルツシルトの最大の特徴は、そのゲームシステムにあります。単なるウォーシミュレーションではなく、ゲームを進めていくにしたがって、次々に新たな目的が現われ、プレイヤーは知らず知らずにゲームのシナリオに引き込まれていくという、ドラマティックなゲーム展開が魅力の、SFシミュレーションゲームです。

究極のゲームシナリオ

ゲームのおもしろさはシナリオで決まります。軍事行動、外交政策調査・研究、資金運用、商業取り引きといった戦略要素を完璧にシミュレート。シミュレーションゲームの面白さを徹底的に追求した究極のゲームシナリオです。

●5"2HD・3枚組¥12,800(価格には消費税は含まれておりません)

■通信販売(送料無料)のお知らせ
工画堂スタジオでは通信販売をしております。ご希望の方は、品名・機種名・住所・氏名・電話番号を明記の上、3%の消費税を加算して現金書留でお申し込みください。

〒162 東京都新宿区市谷台町11
TEL.03-3353-7724

資料請求券
Oh!X・2月号

ATOMIC ROBO-KID アトミック・ロボキッド

ATOMIC R OBO -KID

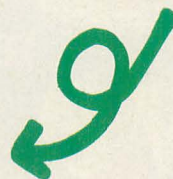
新発売! 全方向スクロールロボットシューティングの極限!!

2×××年、核戦争の難を逃れた少数の人々は、地下に隠れて放射能の嵐がやむのを待ち、地上に出た。しかし、人々のDNA(染色体)は、わずかに残った放射能によって破壊され、人類は自分達の子孫を残せなくなってしまう。トミタ博士は、わずかに残された人々をシェルターに冷凍保存しDNA正常化のプログラムを開発する。ところが、シェルターに向かうロボキッドが動き出す直前に博士は、死んだ。自分の目的も解らずに、目覚めたロボキッド。……はたして、ロボキッドは人類を救う事ができるのか?

アトミックパワーガン

かなり厚い壁まで突き通す程の強力ビームだぜ!
ただし、上下には攻撃出来ましぇ〜ん!

私が Dr.カサイ・ローラー(仮名) である。

ここがポイント
必殺のパワーアップ!!

日ごろ鍛えたこの腕で、みなさんにアドバイスしちゃうぜ。

●ローランド社 MT-32、CM32L、完全対応
MIDIインターフェイスボードC-Z-6BMI
又は、SACOM製SX-68Mが必要です。
標準価格8,800円 copyright ©UPL

3WAY-ビーム

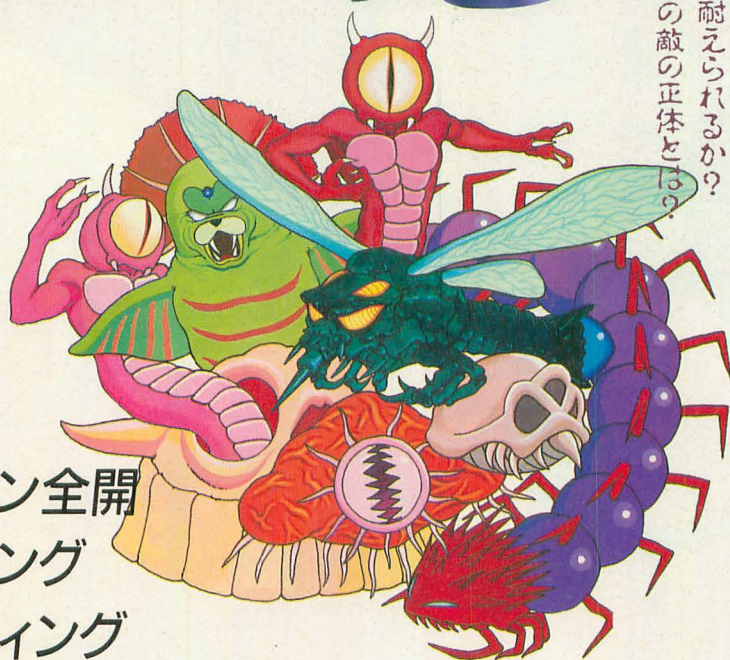
3方向同時のビーム発射がうれしい今日このごろ、だが、弾の威力がそれほど強くないのがたまにキズ。

5WAY-ビーム

編隊で襲ってくるザコキャラにはこれ//
射程距離の短さと威力の弱さに注意するべ〜し!

アトミックミサイル

弾の威力はピカイチ//だが即射ができないのであとは君の腕しだい。

ジェミニウィング
Gemini Wing™

ガンシップ緊急発進(スクランブル)!!

君は迫り来る「蟲」の恐怖に耐えられるか?
そして最後に待ち受ける真の敵の正体とは!?

X68000
アーケードフィールド宣言!! その二

アドレナリン全開
シューティング
ジェミニウィング

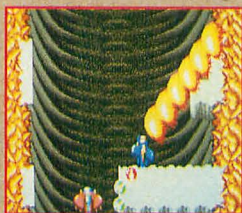
大ヒット驀(ばく)進中!!

好評発売中!!

Check point!

Dr.カサイ・ローラーのワンポイントアドバイス その2

こいつが2面のボス「ツインギドル」だ!

あまり近づかないで、スイングバーで攻撃だ!
ツインギドルの動きが止まったら、
敵のレーザー攻撃の開始だぞ!

これがスイングバーだ
左右に動いて敵をなぎたおす,強力な武器だ。

※標準価格には消費税は含まれておりません。

株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ソフトウェア部 03(3635)7609

未来とは定めら

SIGNA

シグナトリー

構想2年、'91年

3月29日より

X68000 ONL

提供／NCS　制作総指揮・総監督・原作／鈴木 力　脚本／成田伸子　出

制作スタッフ■スクリプト／皆川正三■SE・プログラム／橋谷和幸■メインプログラム／Hかずき■チーフデザイン／

れた運命なのか?

に贈る冒険超大作。

国公開予定!

価¥12,000(5枚組)

ー・フィリップ●バーバラ・ドゥーディ●トーマス・スウェイジ他 制作/Tenky

間繁二郎/大村政幸/矢田智/古澤雅子■音楽・効果音/高橋大昌■NY・南米取材/青空風太郎■NY取材協力/酒井友身

NCS 日本コンピュータシステム株式会社
〒106 東京都港区西麻布4-16-13 第28森ビル TEL.03-3486-6314(代表)
お問い合わせはソフトウェアプロダクト部(直通)03-3486-6588(受付時間/9時～18時)

超低金利クレジットO

注目!!

(平成3年2月末はもちろん)
平成3年3月末一括払い
手数料(金利)無料
ご利用下さい。

またまた秋葉原でおなじみの

●お近くの方は
●本体単品で
●ビジネスソフト

プリンター 10台限定 (送料¥1,000)
■CZ-8PK8(定価¥152,000)
●24ピン漢字プリンタ(136桁)
●ハガキ印字OK!!
P&A限定特価¥49,800
(送料・消費税込¥52,324)

CYBER STICK
●CZ-8NJ2
(定価¥23,800)
超特価!!
¥18,500(送料・消費税込み¥19,570)

X68000シリーズ専用 特価¥16,480
MIDIインターフェースボード
SX-68M(サコム)
(純生コンパチ)定価¥19,800
送料・消費税込み!

X68000用メモリーボード(I/O・DATA)(送料¥500)
①PIO-6BE1-A 定価¥25,000……¥18,000 (送料・消費税込¥19,055)
②PIO-6BE2-2M 定価¥50,000……¥36,000 (送料・消費税込¥38,110)
③PIO-6BE4-4M 定価¥88,000……¥63,000 (送料・消費税込¥66,744)

ジョイスティック 送料¥500
●X-1PRO
定価¥9,500▶特価¥7,800
●ASCII STICK
定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用ください!!

NEW X68000 SUPER/SUPER-HD/PROII/PROII-HD (送料・消費税込)

SUPER
セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

NEW

Ⓐセット:CZ-604C-TN+CZ-606D-TN……定価¥427,800▶特価 価格はTEL下さい。

12回	28,100	24回	14,900	36回	10,300	48回	8,100	60回	6,800

Ⓑセット:CZ-604C-TN+CZ-613D-TN……定価¥483,000▶特価 価格はTEL下さい。

12回	31,800	24回	16,800	36回	11,700	48回	9,100	60回	7,700

SUPER-HD
セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-623C-TN+CZ-606D-TN……定価¥577,800▶特価 価格はTEL下さい。

12回	37,900	24回	20,000	36回	13,900	48回	10,900	60回	9,100

Ⓑセット:CZ-623C-TN+CZ-613D-TN……定価¥633,000▶特価 価格はTEL下さい。

12回	41,600	24回	22,000	36回	15,300	48回	11,900	60回	10,000

PROII
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-653C+CZ-606D……定価¥364,800▶特価 価格はTEL下さい。

12回	22,600	24回	11,900	36回	8,300	48回	6,500	60回	5,400

Ⓑセット:CZ-653C+CZ-605D……定価¥400,000▶特価 価格はTEL下さい。

12回	12,800	24回	13,000	36回	9,000	48回	7,100	60回	5,900

Ⓒセット:CZ-653C+CZ-604D……定価¥798,000▶特価 価格はTEL下さい。
Ⓓセット:CZ-653C+CZ-613D……定価¥420,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-653C+CU-21HD……定価¥433,000▶特価 価格はTEL下さい。

PROII-HD
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-663C+CZ-606D……定価¥474,800▶特価 価格はTEL下さい。

12回	30,800	24回	16,300	36回	11,300	48回	8,800	60回	7,400

Ⓑセット:CZ-663C+CZ-605D……定価¥510,000▶特価 価格はTEL下さい。

12回	32,500	24回	17,100	36回	11,900	48回	9,300	60回	7,800

Ⓒセット:CZ-663C+CZ-604D……定価¥489,800▶特価 価格はTEL下さい。
Ⓓセット:CZ-663C+CZ-613D……定価¥530,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-663C+CU-21HD……定価¥543,000▶特価 価格はTEL下さい。

セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

◎電話にて、ドンドンお問合せ下さい!!
クレジット表には、出せないほどの価格です。
メーカーさん、ご免なさい。
ユーザーの方には大歓迎されそうです。
今がチャンスです、ハイ。

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!

台数限定 送料、消費税込み ※セットでお買い上げの方に、●ディスケット10枚、●ジョイカード2個 プレゼント中!!

EXPERTII

Ⓐセット:P&A厳選セット
■CZ-603C
(本体価格¥338,000)
+
■CZ-606D
(モニター定価¥79,800)
▶P&A超特価 ¥304,000

Ⓑセット
■CZ-603C+CZ-604D
定価¥432,800…▶特価¥309,000

Ⓒセット:
■CZ-603C+DZ-605D
定価¥453,000…▶特価¥322,000

Ⓓセット:
■CZ-603C+CZ-613D
定価¥473,000…▶特価¥342,000

Ⓔセット:
■CZ-603C+CU-21HD
定価¥486,000…▶特価¥347,000

EXPERT-HD

Ⓐセット:P&A厳選セット
■CZ-612C(ブラック)
(本体価格¥466,000)
+
■CZ-606D(ブラック)
(モニター定価¥79,800)
▶P&A超特価 ¥335,000

Ⓑセット:
■CZ-612C+CZ-604D
定価¥560,800…▶特価¥340,000

Ⓒセット:
■CZ-612C+CZ-605D
定価¥581,000…▶超特価¥359,000

Ⓓセット:
■CZ-612C+CZ-613D
定価¥601,000…▶超特価¥372,000

Ⓔセット:
■CZ-612C+CU-21HD
定価¥614,000…▶超特価¥386,000

■NEC=モデム(定価¥44,800)
⦿COMSTARZ 2424/5
●2400/1200bps全二重
●MNP5クラス
●インターフェース付

P&A超特価
¥28,500
(送料・消費税込み¥30,385)

■ALL in Note
フリートップ
パーソナルコンピュータ
⦿AX-286N-H2
(定価¥398,000)
P&A超特価
価格はTEL下さい。

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00~PM7:00、日祭AM10:00~PM6:00

～84回払いまでOK‼

★頭金なし！★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕‼

寄り下さい。専門係員が説明いたします。
で受付します。詳しくは電話にてお問合せ下さい。
の20%引きOK！ TELください。

全国通販

超特価でクレジットが組める‼

X68000用ソフトコーナー（送料1ヶ～5ヶまで￥500）

商品	定価	特価
Z's STAFF PRO68K Ver2.0（ツァイト）	￥58,000	￥39,500
Z's TRIPHONY デジタルクラフト（ツァイト）	￥39,800	￥27,500
テラッツォ（ハミングバード）	￥19,400	~~￥15,840~~
KAMIKAZE（サムシング・グッド）	￥68,800	￥45,500
C&Professional Pack（マイクロウェアジャパン）	￥58,000	￥43,000
Final Ver3.2（エーエスピー）	￥38,000	~~￥30,400~~
C-compiler PRO68K Ver.2 CZ-245L	￥44,800	￥34,500
CARD PRO68K CZ226BS	￥29,800	TEL下さい。
C compiler PRO68K CZ211LS	￥39,800	￥28,500
OS-9/X68000 CZ219SS	￥29,800	￥20,700
AI-68K CZ234LS	￥188,000	TEL下さい。
THE 福袋V2.0 CZ224LS	￥9,900	￥7,400
SOUND PRO68K	￥15,800	￥11,300
MUSIC PRO68K CZ213MS	￥15,800	￥13,300
Sampling PRO68K CZ215MS	￥17,800	￥12,500
MUSIC-studio PRO68K 237MS	￥15,800	TEL下さい。
MUSIC-PRO68K〔MIDI〕247MS	￥28,800	￥20,500
New-print Shop 221HS	￥19,800	TEL下さい。
Communication 223CS	￥19,800	￥9,800
Communication Ver.2 CZ-257CS	￥19,800	￥15,500
C-TRACE68 Ver.3.0（キャスト）	￥98,000	￥69,800
サイクロンEXPRESSα68	￥98,000	TEL下さい。
G68K Ver2 PRO	￥22,000	~~￥17,600~~
THE FILE PROFESSOR（ロゴシステム）	￥28,000	~~￥22,400~~
Gツール（ザインソフト）	￥28,000	~~￥22,400~~
たーみのる2（SPS）	￥17,800	~~￥14,200~~
マジックパレット（ミュージカルプラン）	￥19,800	~~￥15,800~~
Hyper word CZ-251BS	￥39,800	~~￥31,800~~

●ゲームソフト20%OFF OK‼（一部ソフト除く）

周辺機器コーナー（送料￥500）

商品	定価	特価
Ⓐ CZ-8NSI	￥188,000	￥145,000
Ⓑ CZ-6VTI	￥69,800	￥54,000
Ⓒ CZ-6TU	￥33,100	￥25,000
Ⓓ BF-68PRO	￥19,800	￥15,500
Ⓔ CZ-6BEI	￥35,000	￥26,500
Ⓕ CZ-6BEIA	￥38,000	￥28,600
Ⓖ CZ-6BE2	￥79,800	￥60,000
Ⓗ CZ-6BE4	￥138,000	￥107,000
Ⓘ CZ-6BFI	￥49,800	￥38,200
Ⓙ CZ-6BPI	￥79,800	￥61,000
Ⓚ CZ-6BMI	￥26,800	￥20,300
Ⓛ CZ-6EBI	￥88,000	￥67,500
Ⓜ AN-S100	￥36,600	￥28,500
Ⓝ CZ-6SDI	￥44,800	￥35,000
Ⓞ CZ-8PC3	￥65,800	P&A超特価TEL下さい。
Ⓟ CZ-8PC4	￥99,800	P&A超特価TEL下さい。
Ⓠ CZ-8PG1	￥130,000	P&A超特価TEL下さい。
Ⓡ CZ-8PG2	￥160,000	P&A超特価TEL下さい。
Ⓢ CZ-8PK10	￥97,800	P&A超特価TEL下さい。
Ⓣ CZ-6PVI	￥198,000	￥153,000
Ⓤ IO-735X	￥248,000	￥190,000
Ⓥ CZ-8BSI	￥23,800	￥19,000
Ⓦ PIO-6BE1-A（I/O DATA）	￥25,000	￥18,000
Ⓧ PIO-6BE2-2M（I/O DATA）	￥50,000	￥36,000
Ⓨ PIO-6BE4-4M（I/O DATA）	￥88,000	￥63,000

X68000用ハードディスク（送料￥1,000）

アイテム
- ●HXD-040（40MB/23ms）…定価￥118,000▶特価￥88,000
- ●HXD-042（増設用）…定価￥128,000▶特価￥95,000

アイテック
- ●ITX-640（40MB/28ms）…定価￥158,000▶特価￥83,000
- ●ITX-680（80MB/20ms）…定価￥198,000▶特価￥97,000

プリンター（ケーブル・用紙付）限定5台　新品（送料￥1,000）

- ●CZ-8PC3（カラー漢字24ドット熱転写プリンター）定価￥65,800…特価￥39,800
- ●CZ-8PK8（24ピン漢字プリンター136桁）定価￥152,000…特価￥49,800
- ●CZ-8PC4 P&A特選‼（カラー漢字48ドット熱転写プリンター）定価￥99,800…特価￥57,000

モデムコーナー　（送料￥1,000）

商品	定価	特価
Ⓐ MD-24FS5（オムロン）	￥49,800	￥34,800
Ⓑ MD-24FS7（オムロン）	￥64,800	￥45,000
Ⓒ コムスター2424/4（NEC）	￥38,800	￥28,000
Ⓓ コムスター2424/5（NEC）	￥44,800	~~￥28,500~~

P & A 特選パソコンラック　（送料無料）移動自由（キャスター付）

中古パソコン（セットはモニター付）送料￥2,000

- ●X68000セット…▶￥180,000
- ●X68000 ACEセット…▶￥200,000
- ●X68000 ACE-HDセット…▶￥215,000
- ●EXPERTセット…▶￥230,000
- ●EXPERT-HDセット…▶￥265,000
- ●PROセット…▶￥250,000
- ●X68000PRO-HDセット…▶￥270,000
- ●EXPERTⅡセット…▶￥250,000
- ●EXPERTⅡ-HDセット…▶￥320,000
- ●PROⅡセット…▶￥240,000
- ●PROⅡ-HDセット…▶￥310,000

中古パソコンはP&Aにおまかせ‼

その場で高価現金買取り・高価下取りOK‼

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- ●月々￥1,000円からOK‼
- ●ボーナス払いOK（夏冬10回までOK）
- ●支払い回数 1回～84回
- ●お支払いは、8ヶ月先からでもOK‼

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

〔振込先〕住友銀行　新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.5	4.5	6.0	6.0	11.0	12.5	17.5	23.0	29.5	38.0	45.5

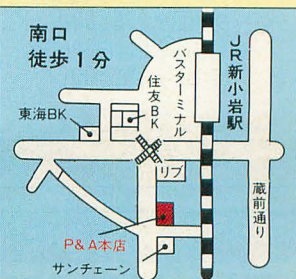
アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます（祭日の場合は翌日になります）

南口 徒歩1分
東海BK　住友BK　バスターミナル　JR新小岩駅　リブ　蔵前通り　P&A本店　サンチェーン

●マイコン
●ビデオ
●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148（代）FAX.03-3651-0141

営業時間
平日：AM10：00～PM7：00
日祭：AM10：00～PM6：00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利オクトハッピークレジットをご利用下さい!!

■朗報です。平成3年2月末一括払いOK!!手数料なし。ご利用下さい。■店頭にて、ゲームソフト25%OFF!!

案内図

至品川　東口ロータリー　JR蒲田から徒歩5分　京浜蒲田から徒歩1分　至品川
三菱BK　JR蒲田駅　2F　商店街アーケード　京浜蒲田駅
富士BK　三和BK　メガネドラッグ　八幡神社
環八通り　NTT　〒　至横浜　至横浜

店頭セール実施中

――オクトで始まるパソコンワールド――

☎03-3730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-3730-6271

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

1回	2.06	3回	3	6回	4	10回	5.5	12回	5.5	15回	8	18回	10
20回	11	24回	12	30回	16	36回	17	48回	22	60回	28		

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回～60回)頭金ナシOK!
- ▶ボーナス一括払いOK!ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

蒲田

平成3年2月末一括払いOK!!
新春恒例お年玉セール実施中。
▶今月のセットは、超お買徳!!電話で交渉すべし!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●① MD-2HD 10枚 ②ジョイカード 2個(連射式) ③シリコンキーボードカバー ④ゲームソフト サンダーブレード(¥9500)

お好みのセットをお選び下さい。送料無料!!

●CZ-604C-TN
定価¥348,000

現金特価!! 推選 お電話下さい。

● SX-WINDOW搭載。
●拡張I/Oポート4スロット装備

PROII・PROII-HD
● CZ-653C-BK/GY
定価¥285,000
● CZ-663C-BK/GY
定価¥395,000

CZ-8NJ2 限定
●インテリジェントコントローラ
定価¥23,800
超特価¥18,000

15型カラーディスプレイTV

CZ-605D-GY/BK
定価¥115,000

15型カラーディスプレイTV

CZ-613D-GY/BK
定価¥135,000

14型カラーディスプレー

CZ-606D(GY/BK/TN)

21型カラーディスプレイ

CU-21HD
定価¥148,000

Ⓐ CZ-604C＋CZ-605D…・定価合計¥463,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑ CZ-653C＋CZ-605D…・定価合計¥400,000▶オクト大特価

12回	¥24,600	24回	¥13,000	36回	¥9,100	48回	¥7,100	60回	¥5,900

Ⓒ CZ-663C＋CZ-605D…・定価合計¥510,000▶オクト大特価

12回	¥33,500	24回	¥17,700	36回	¥12,300	48回	¥9,600	60回	¥8,100

Ⓓ CZ-604C＋CZ-613D…・定価合計¥483,000▶オクト大特価

12回	¥31,800	24回	¥16,900	36回	¥11,700	48回	¥9,200	60回	¥7,700

Ⓔ CZ-653C＋CZ-613D…・定価合計¥420,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓕ CZ-663C＋CZ-613D…・定価合計¥530,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓖ CZ-604C＋CZ-606D…・定価合計¥427,800▶オクト大特価

12回	¥28,000	24回	¥14,900	36回	¥10,300	48回	¥8,100	60回	¥6,800

Ⓗ CZ-653C＋CZ-606D…・定価合計¥364,800▶オクト大特価

12回	¥22,400	24回	¥11,900	36回	¥8,300	48回	¥6,400	60回	¥5,400

Ⓘ CZ-663C＋CZ-606D…・定価合計¥474,800▶オクト大特価

12回	¥31,200	24回	¥16,500	36回	¥11,500	48回	¥9,000	60回	¥7,500

Ⓙ CZ-604C＋CU-21HD…定価合計¥496,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓚ CZ-653C＋CU-21HD…定価合計¥433,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓛ CZ-663C＋CU-21HD…定価合計¥543,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?	60回	?

♥本体セットは、1ヶ月間(1/15～2/14)だけの大特価セール!!
♥クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料 (注)本体セット以外の周辺機器(プリンター、モデム、HDD等)及びソフトの送料は、北海道・九州地区＝1ケロ¥1500、■その他離島地区は、1ケロ¥2000となります。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

オクト厳定!! SUPER-HDスペシャルセット～今がチャンス!! 電話で値切ろう(送料無料)

●ザ・ワークステーションと呼ぶにふさわしい

SUPER-HD＝オクト厳定セット!!

※マウス・トラックボール付!!
ディスプレイにはスピーカ2個
チルト台付!!

Ⓐ：CZ-623C-TN＋CZ-606D-TN…………定価合計￥577,800▶大特価

12回	￥37,800	24回	￥20,000	36回	￥13,900	48回	￥10,900	60回	￥9,100

Ⓑ：CZ-623C-TN＋CZ-613D-TN…………定価合計￥633,000▶大特価

12回	￥41,600	24回	￥22,000	36回	￥15,300	48回	￥12,000	60回	￥10,100

現金特価!!
お電話下さい。

上記セットお買い上げの方に、オクトからのプレゼント!!
①MD-2HD 10枚
②サンダーブレード(ゲーム￥9,500)
③ジョイカード(連射式)
④シリコンキーボードカバー(￥2,800)

※超低金利クレジットをご利用下さい。1回～60回払い、頭金ナシ!! ボーナス1回及び2回払いOKです。

■店頭大恩謝セール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!!

X68000ソフト大セール実施中※ゲームソフトオール25%off 送料￥500

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0
(シャフト)定価￥58,000……特価￥39,800

〈グラフィック〉●デジタルクラフト
定価￥39,800……特価￥28,000

〈データーベース〉●KAMIKAZE
定価￥68,000……特価￥45,800

〈ワープロ〉●ハイパーワード
定価￥39,800 CZ-251BS……特価￥29,800

〈グラフィック〉●C-TRACE68
(キャスト)定価￥68,000……特価￥51,000

〈開発ツール〉●C-コンパラPRO68KV.2
定価￥44,800 CZ-245IS……特価￥33,600

〈C言語〉●C&Professional Pack
定価￥58,000……特価￥43,800

〈CGツール〉●CANVAS PRO68K
定価￥29,800 CZ-249GS……特価￥22,400

〈グラフィック〉●サイクロン エキスプレス
定価￥78,000……特価￥28,000

型名	商品	定価	特価
CZ-212BS	BUSINESS PRO-68K	￥68,000	￥48,000
CZ-213MS	MUSIC PRO68K	￥18,800	￥13,500
CZ-214MS	SOUND PRO-68K	￥15,800	￥11,500
CZ-215MS	Sampling PRO-68K	￥17,800	￥12,800
CZ-219SS	OS-9/X68000	￥29,800	￥21,000
CZ-220BS	DATA PRO-68K	￥58,000	￥41,000
CZ-223CS	Communication PRO-68K	￥19,800	￥14,300
CZ-224LS	THE 福袋 V2.0	￥9,900	￥7,500
CZ-226BS	CARD PRO-68K	￥29,800	￥21,300
CZ-241BS	システム手帳リフィル集	￥9,800	￥7,500
CZ-242BS	活用フォーム集	￥9,800	￥7,500
CZ-244SS	Human 68K Ver2.0	￥9,800	￥7,500
CZ-247MS	MUSIC PRO-68K(MIDI)	￥28,800	￥20,800
CZ-240BS	Stationery PRO-68K	￥14,800	￥11,500
CZ-243BS	CYBER NOTE PRO-68K	￥19,800	￥15,200
EW		￥38,000	￥29,800
G-68K		￥14,800	￥11,400
E-68		￥19,800	￥15,300
CZ-255GS	CANVASドローグラフィックLIB	￥8,800	￥6,600
CZ-256GS	CANVASドローグラフィックVOL.2	￥8,800	￥6,600
CZ-260LS	XBAS to CHECKER PRO68K	￥9,800	￥7,500
CZ-259SS	SX-WINDOW Ver.1.0	￥6,800	￥5,000
CZ-234LS	AI-68K	￥188,000	￥139,000
CZ-252MS	MUSIC STUDIO PRO68K	￥28,800	￥21,500

モデムコーナー (送料￥1,000)

オムロン
●MD-1200AIII……特価￥14,500
●MD-12FS……特価￥15,000
●MD-24FP4II……特価￥26,000
●MD-24FN5……特価￥30,000
●MD-24FS4……特価￥31,000
●MD-24FS5……特価￥34,000
●MD-24FS7……特価￥44,000
●MD-24FC5……特価￥34,000
●MD-24FP5II……特価￥28,500
●MD-24FN4……特価￥27,000
●MD-24FJ4……特価￥31,000
●MD-24FJ5……特価￥34,000
●MD-24HS……特価￥64,000
●MD-48HS……特価￥98,000
●MD-96FS5……特価￥131,000

AIWA
●PV-A24VM5……特価￥30,000
●PV-M24VM5……特価￥30,000
●PV-A12……特価￥14,500
●PV-M24……特価￥28,500

熱転写カラー漢字プリンター (ケーブル用紙付) 送料￥1,000

①CZ-8PK10(24ピン漢字プリンター136桁)
定価￥97,800……大特価!! TEL下さい。
②CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価￥130,000……大特価!! TEL下さい。
③CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価￥160,000……大特価!! TEL下さい。
④IO-735×(カラーイメージシェット)
定価￥248,000……大特価!! TEL下さい。

■IO-735X (カラーイメージジェットプリント)
大特価TEL下さい

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる棚板5段のマルチに活用できるディスク。ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価￥15,500

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!使い易いデスクです。
1245(H)×614(W)×600(D)
特価￥11,500

③三段キャスター付

3段キャスター付
場所を選ばない簡易で便利なディスクです。
限定
1175(H)×640(W)×600(D)
特価￥8,800

周辺機器コーナー (送料￥1,000)

●CZ-6BE1 IBM増設RAMボード……(￥35,000)▶特価￥26,500
●CZ-6BE1B IMB増設RAMボード……(￥28,000)▶特価￥21,000
●CZ-6BE2 2MB増設RAMボード……(￥79,800)▶特価￥60,500
●CZ-6BE4 4MB増設RAMボード……(￥138,000)▶特価￥104,800
●CZ-6BF1 増設用RS-232Cボード……(￥49,800)▶特価￥38,500
●CZ-6BG1 GP-IBボード……(￥59,800)▶特価￥45,000
●CZ-6BM1 MIDIボード……(￥26,800)▶特価￥20,500
●CZ-6BN1 スキャナ用パラレルボード……(￥29,800)▶特価￥22,800
●CZ-6BP1 数値演算プロセッサボード……(￥79,800)▶特価￥60,500
●CZ-6BO1 ユニバーサルI/Oボード……(￥39,800)▶特価￥30,500
●CZ-6EB1/BK 拡張I/Oボックス……(￥88,000)▶特価￥66,800
●CZ-6VT1/BK カラーイメージ・ユニット……(￥69,800)▶特価￥53,000
●CZ-8NM2A マウス……(￥68,800)▶特価￥5,300
●CZ-8NT1 マウストラックボール……(￥98,800)▶特価￥7,500
●CZ-8NS1 カラーイメージスキャナ……(￥188,000)▶大特価
●CZ-6BC1 FAXボード……(￥79,800)▶特価￥60,500
●CZ-8TM2 モデムユニット……(￥49,800)▶特価￥38,000
●CZ-64H 増設ハードディスク……(￥120,000)▶大特価
●CZ-6TU GY/BK RGBシステムチューナー……(￥33,100)▶特価￥25,000
●BF-68PRO 高性能CRTフィルター……(￥19,800)▶特価￥15,500
●CZ-6MO1 光磁気ディスクユニット……(￥450,000)▶大特価
●CZ-6BS1 SCSIインターフェースボード……(￥29,800)▶特価￥22,400
●CZ-6BL2 LANボード……(￥298,000)▶大特価

特選周辺機器(送料￥1,000)

●SX-68M MIDインターフェースボード
(システムサコム)￥19,800……特価￥15,000
●CZ-6BV1 ビデオボード
￥21,000……特価￥15,700

■増設RAMボード＝I・Oデータ
①PIO-6BE1-A(1MB)
￥25,000……特価￥18,000
②PIO-6BE2-2M(2MB)
￥50,000……特価￥36,300
③PIO-6BE4-4M(4MB)
￥88,000……特価￥64,000

店頭ゲームソフトオール25%off! ビジネスソフト25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2.06%	3回	3%	6回	4%	10回	5.5%	12回	5.5%
15回	8%	18回	10%	20回	11%	24回	12%	30回	16%
36回	17%	48回	22%	60回	28%				

振込先
富士銀行 久ヶ原支店 ㊔No.1824
三菱銀行 蒲田支店 ㊔No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-3770-8855

■アフターサービス万全のサポート体制
- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…。午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

■TEL・FAXのお見積OK.!!
■低金利クレジットをご利用下さい。

▶1・15〜2・14

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2 (定価¥23,800)
OAランド特価 ▶¥18,000

電子手帳 ●見やすい漢字4桁表示!! 情報任時代の必需品!!
- ■PA-9500(¥48,000)…▶特価¥38,000
- ■PA-8500(¥28,000)…▶特価¥15,000
- ■PA-7500(¥22,000)…▶特価¥12,000

SHARPのことならなんでもおまかせ!!

大徳買セール! 安く値切ってネ。
お電話下さい。㊙価格をお知らせいたします。

SHARP X68000シリーズセット どんどんTEL下さい。

X68000 SUPER NEW

①CZ-604C-TN+CZ-613D-TN
定価合計¥483,000

1回	355,000	12回	32,100
24回	16,900	36回	11,700

②CZ-604C-TN+CZ-606D-TN
定価合計¥427,800

1回	315,000	12回	28,400
24回	15,000	36回	10,400

■CZ-604C 特価~~¥348,000~~
■CZ-623C 特価~~¥498,000~~

X68000 SUPER-HD

①CZ-623C-TN+CZ-613D-TN
定価合計¥633,000

1回	466,000	12回	42,100
24回	22,200	36回	15,400

②CZ-623C-TN+CZ-606D-TN
定価合計¥577,800

1回	425,000	12回	38,400
24回	20,300	36回	14,000

X68000 EXPERT-II

①CZ-603C+CZ-613D
定価合計¥473,000

1回	348,000	12回	31,400
24回	16,600	36回	11,500

②CZ-603C+CZ-605D
定価合計¥453,000

1回	333,000	12回	30,100
24回	15,900	36回	11,000

③CZ-603C+CZ-606D
定価合計¥417,800

1回	TEL下さい。	12回	27,800
24回	14,600	36回	10,100

■CZ-603C 特価~~¥249,000~~
■CZ-613C 特価~~¥448,000~~

X68000 EXPERT II-HD

①CZ-613C+CZ-613D
定価合計¥583,000

1回	TEL下さい。	12回	38,800
24回	20,500	36回	14,200

②CZ-613C+CZ-605D
定価合計¥563,000

1回	414,000	12回	37,400
24回	19,500	36回	13,700

③CZ-613C+CZ-606D
定価合計¥527,800

1回	TEL下さい。	12回	35,100
24回	18,500	36回	12,800

X68000 PROII

①CZ-653C+CZ-613D
定価合計¥420,000

1回	288,000	12回	26,000
24回	13,700	36回	9,500

②CZ-653C+CZ-605D
定価合計¥400,000

1回	273,000	12回	24,700
24回	13,000	36回	9,000

③CZ-653C+CZ-606D
定価合計¥364,800

1回	TEL下さい。	12回	22,300
24回	11,800	36回	8,200

■CZ-653C 特価~~¥285,000~~
■CZ-663C 特価~~¥395,000~~

X68000 PROII-HD

①CZ-663C+CZ-613D
定価合計¥530,000

1回	TEL下さい。	12回	35,200
24回	18,600	36回	12,900

②CZ-663C+CZ-605D
定価合計¥510,000

1回	TEL下さい。	12回	33,900
24回	17,900	36回	12,400

③CZ-663C+CZ-606D
定価合計¥474,800

1回	TEL下さい。	12回	31,600
24回	16,600	36回	11,500

■期間中、セットでお買い上げの方には、①Vボール②ニュージーランド・ストーリー(ゲーム)のがついてきます。さらに、③テトリスやドルアーガの塔などの入ったゲームパックもプレゼント.!!

周辺機器

- ■光磁気ディスクユニット ●CZ-6MO1 (定価¥450,000) 特価¥335,000
- ■SCSIボード ●CZ-6BS1 (定価¥29,800) 特価¥22,300
- ■ビデオボード ●CZ-6BV-1 (定価¥21,000) 特価¥15,600

周辺機器コーナー 電話で値切ろう。

プリンターセットコーナー

①CZ-8PC4(GY)(48ドット/カラー対応/ハガキ可能)
定価¥99,800……特価~~¥64,800~~

②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……特価¥73,800

③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥96,000

④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥117,800

OAランド特選品.!!

■IO-735X(定価¥248,000)
●カラーイメージジェットプリンター ●ケーブル付
特価¥186,000

モデム

オムロン	MD-1200A III	¥14,500
	MD-24FP4 II	¥27,500
	MD-24FP5 II	¥29,800
	MD-24FN4	¥28,000
	MD-24FN5	¥31,300
	MD-24FJ4	¥31,300
	MD-24FJ5	¥34,500
	MD-24FS4	¥28,500
	MD-24FS5	¥34,500
アイワ	PV-A24VM5	¥32,500
	PV-M24	¥28,800
NEC	COMSTAR 2424/4	¥28,800
	COMSTAR 2424/5	¥33,500

X68000用周辺機器コーナー

①CZ-6VT1(カラーイメージユニット)
定価¥69,800……特価¥52,500
②CZ-8NS1(カラーイメージスキャナー)
定価¥88,000……特価¥138,000
③CZ-6BM1(MIDIボード)
定価¥26,800……特価¥20,500
④CZ-8NJ2(インテリジェント・コントローラー)
定価¥23,800……特価¥18,000
⑤CZ-6TU(RGBシステムチューナー)
定価¥33,100……特価¥25,000
⑥CZ-64H(増設ハードディスク)
定価¥120,000……特価¥89,000
⑦CZ-6EB1(拡張I/Oボックス=4スロット)
定価¥88,000……特価¥66,000
⑧CZ-6BP1(数値演算プロセッサボード)
定価¥79,800……特価¥60,000

I・Oデータ増設RAMボード

■PIO-6BE2-2M(2MB)
定価¥50,000
特価¥35,900

■PIO-6BE1-A(1MB)
定価¥25,000
特価¥18,800

■PIO-6BE4-4M(4MB)
定価¥88,000
特価¥62,800

《計測技研》

●高速増設メモリと数値演算プロセッサが一つのボードになった!!

型番	定価	特価
KGB-X68PRK-00	(¥34,000)	特価¥27,000
-01	(¥58,000)	特価¥46,000
-02	(¥74,000)	特価¥59,000
-03	(¥98,000)	特価¥78,000
-04	(¥122,000)	特価¥97,000
-10	(¥72,000)	特価¥57,000
-11	(¥96,000)	特価¥76,000
-12	(¥112,000)	特価¥89,000
-13	(¥136,000)	特価¥108,000
-14	(¥160,000)	特価¥127,000

OAランド今月の大目玉!!=超A級中古品

今ガチャンス!!

- ●CZ-603C-GY……特価¥200,000
- ●CZ-613D-GY……特価¥79,000
- ●CZ-603C-BK……特価¥218,000 (メーカー保証付)
- ●CU-21HD-BK……特価¥87,000 (メーカー保証付)
- ●CZ-8PK8(2台)……特価¥40,000
- ●CZ-8PK9……特価¥38,000
- ●CZ-8NS1……特価¥113,000
- ●CZ-6BN1……特価¥12,000 (パラレルボード)
- ●KGB-X68PRK-00……特価¥22,000 (メモリ&コプロ増設ボード)

OAランド推奨ソフト

- ■SX-WINDOW (次世代インテリジェントソフト) 定価¥6,800 特価¥5,100
- ■CZ-245LS (C-コンパイラII) 定価¥44,800 特価¥33,500
- ■CZ-260LS (X Bas to C CHECKER) 定価¥9,800 特価¥8,000
- ■CZ-249GS (CANVAS-PRO68K) 定価¥29,800 特価¥22,300
- ■CZ-255GS/256GS (ドローグラフィックライブラリ1/2) 定価¥8,800 特価¥7,000
- ■CZ-219SS (OS9/X68,000) 定価¥29,800 特価¥23,800

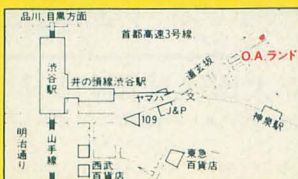
通信販売のご案内

——全国通販——

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

品川、目黒方面 首都高速3号線 渋谷駅 井の頭線渋谷駅 ヤマハ J&P 109 神泉駅 明治通り 山手線 西武百貨店 東急百貨店 O.A.ランド

■年中無休です.!!

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

クレジット表

3回	3%	6回	4%	10回	5.5%	12回	5.5%	15回	8%	18回	10%	20回	11%
24回	11.5%	30回	15.5%	36回	16%	42回	20.5%	48回	21%	54回	26.5%	60回	27%

㈱オー・エー・ランド

〒150東京都渋谷区円山町20-4 第5日新ビル1F
☎(03)3770-8555

関東エリアの送料は、1個につき¥1,000です。 FAX(03)3770-7080

★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、12月下現在です。

ツクモ全店史上最大の決算セール 1/18～1/30迄

掲載商品代金2万円以上送料無料　とにかく一度、御来店下さい！

世界一広い、ツクモパソコン本店 X68000コーナーでお客様のおいでをお待ちしております!!

そこで移転に伴い古くなった7号店の展示品の大処分セールを実施します。どれも1、2台の数しかありませんので、ご了承くださいね。

例]他にもありますので是非ご来店ください。

- X68000 EXPERT(CZ-602C-GY)……¥210,000より
- X68000 PRO-HD(CZ-662C)……¥215,000より
- カラーイメージスキャナー(CZ-8NS1)……¥128,000より
- 48ドット熱転写プリンタ(CZ-8PC4-GY)……¥59,800より
- 136桁ドットプリンタ(CZ-8PK8)……¥59,800より
- 20MBハードディスク……¥59,800より
- その他棚ずれ品など
- チルト台(CZ-6ST1E) 台数少……¥2,980より
- MZ-2500ランゲージシリーズソフト……¥1,980より

名古屋、札幌でも決算大セールを行っています。是非、ご来店下さい！

名古屋2号店 横山 ☎052(251)3399

ツクモ札幌 田口 ☎011(241)2299

X68000 シリーズ本体

- PROII CZ-653C 定価¥285,000
- PROII-HD CZ-663C 定価¥395,000
- EXPERTII CZ-603C 定価¥338,000
- EXPERTII-HD CZ-613C 定価¥448,000
- SUPER NEW CZ-604C 定価¥348,000
- SUPER-HD CZ-623C 定価¥498,000

ツクモ特価販売中！

メモリーボード

1MB増設RAMボード (ACE/PROシリーズ用) (消費税別途¥570) ツクモ特価¥19,000

2MB増設RAMボード ツクモ特価¥37,000 (消費税別途¥1,110)

4MB増設RAMボード ツクモ特価¥64,000 (消費税別途¥1,920)

※計測技研のメモリーボードも取り扱っております。価格については、お尋ねください。

パワーアップアイテム

ハードディスク

標準タイプハードディスク(SASI)

アイテック IT X640(40MB) ツクモ特価¥84,800(消費税別途¥2,544)

アイテック IT X680(80MB) ツクモ特価¥99,800(消費税別途¥2,994)

SCSIハードディスク

SCSIボード CZ-6BS1 定価¥29,800

IT X80S(80MB) 定価¥128,000 ツクモ特価¥99,800(消費税別途¥2,994)

IT X130S(130MB) 定価¥158,000 ツクモ特価¥125,000(消費税別途¥3,074)

※X68000 SUPERシリーズ以外の機種はCZ-6BS1(SCSIボード)が必要です。

X680 (カラー:ブラック グレー)

IT X130S

光磁気ハードディスク

CZ-6MO1 定価¥450,000 ツクモ特価販売中

※光磁気ディスクカートリッジは別売です。(30,000円)

※X68000 SUPERシリーズ以外の機種はCZ-6BS1(SCSIボード)が必要です。

CZ-6MO1

SONY 光磁気ディスクユニットセット

- NWP-539N 光磁気ディスクユニット
- CZ-6BS1 SCSIボード
- SCSIケーブル
- 光磁気ディスクカートリッジ

合計定価¥509,800

ツクモ特価¥398,000 (消費税別途¥11,940)

クレジット例(42回払・税込) 初回¥14,274＋月々¥12,300×41回

アートツール

■ハードウェア

- 一流メーカー イメージスキャナ 台数限定……ツクモ特価¥128,000(消費税別途¥3,840)
- CZ-6BV1 ビデオボード……定価¥21,000
- CZ-6VT1 カラーイメージユニット……定価¥69,800
- CZ-6BP1 数値演算プロセッサボード……定価¥79,800

■ソフトウェア

- CANVAS PRO-68K……定価¥29,800
- Z's STAFF PRO-68K……ツクモ特価¥49,300(消費税別途¥1,479)
- マジックパレット……ツクモ特価¥16,800(消費税別途¥504)
- 彩クロンExpressα68……ツクモ特価¥83,300(消費税別途¥2,499)
- デジタルクラフト……ツクモ特価¥33,800(消費税別途¥1,014)

開発ツール

- C Compiler PRO-68K Ver2.0……定価¥44,800
- SX-WINDOW……定価¥6,800
- XBAS to C CHECKER PRO-68K……定価¥9,800

ミュージックツール ＝NEWプライス＝

お手軽なMIDセットA

- CM-32L……定価¥69,000
- SX-68M……定価¥19,800
- Musicstudio Mu-1 Ver1.4 定価¥19,800

ツクモ特価 ¥88,000 (消費税別途¥2,640)

クレジット例(18回払・税込) 初回¥7,223＋月々¥5,600×17回

お手軽なMIDIセットB

- CM-64……定価¥129,000
- SX-68M……定価¥19,800
- Musicstudio Mu-1 Ver1.4 定価¥19,800

ツクモ特価 ¥138,000 (消費税別途¥4,140)

クレジット例(24回払・税込) 初回¥7,603＋月々¥6,900×23回

※「Musicstudio PRO68K Ver2.0又は「Music PRO68K」<MIDI>のソフトの場合には、¥9,500プラスになります。また、これらのソフトウェアがバージョンアップにより価格が変更になった場合には変更となります。

追加オプション機器

はなうたくん CP-40 定価¥33,000

MIDIキーボードコントローラー PC-200 定価¥36,000

情報ツール

電子手帳シリーズ

ハイパー電子システム手帳 PA-9500 新発売 定価¥48,000

- 表計算カード PA-9C1……定価¥16,000
- RAMカード PA-9C90 (64Kbyte) 定価¥14,000
- PA-9C91 (128Kbyte) 定価¥20,000

PA-8800……ツクモ特価¥24,800 (消費税別途¥744)

CE-300L 通信ケーブル……ツクモ特価¥2,500 (消費税別途¥75)

電子手帳対応ソフト

- CYBER NOTE PRO-68K……定価¥19,800
- Stationery PRO-68K……定価¥14,800

ツクモグローバルカード

大/好/評/入/会/者/募/集

国内・外で大活躍！

使って安心、持ってて便利！ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK！

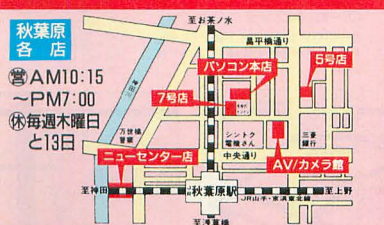
(03)3251-9898又は各店で！

18才以上なら学生でもOK！

各種新製品大特価販売中！詳しくはお問い合せ下さい。

ツクモ通販センター フリーダイヤル受注専門 0120-377-999

商品についてのお問い合せは各店店頭又は☎03(3251)9911へ

秋葉原各店 営AM10:15～PM7:00 休毎週木曜日と13日

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

九十九電機(株) 〒101-91 東京都千代田区神田郵便局私書箱135号

パソコン本店 荒井　N・C店 福地

ツクモパソコン本店 ☎03-3253-5599 (担当/荒井)

便利で安心な通信販売 ツクモ通販センター☎03-3251-9911

- ■ツクモAV/カメラ館 ☎03-3254-3999(担当/川名)
- ■ツクモニューセンター店 ☎03-3251-0987(担当/福地)
- ■ツクモ5号店 ☎03-3251-0531(担当/森)
- ■名古屋1号店 ☎052-263-1655(担当/吉高)
- ■名古屋2号店 ☎052-251-3399(担当/横山)
- ■ツクモ札幌 ☎011-241-2299(担当/田口)

★表示価格には消費税は含まれておりません。

★商品のご注文は在庫確認の上お願いします。

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話1本！配達日の指定もできます。	月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 ツクモデンキ	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

ワールドインアオヤマのコンセプトショップ 池袋本店 1/17 TUE OPEN!

● TOKYU HANDS となり AMLUX 前

AOYAMA FOR THE EVOLUTION OF YOU

X68000 PRO II

X68000PRO II X68000 新Bコース

CZ-653C(本体)……¥285,000
CZ-604D(0.31ステレオスピーカ付ディスプレー)……¥94,800
CZ-8NJ2(インテリジェントコントローラー)……¥23,800
ナイトアームズ……¥9,700
サンダーブレード……¥9,500
スーパーハングオン……¥7,800
パックマンア……¥7,800
ジェノサイド……¥8,800
ヴァリスII……¥9,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥475,000 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000PRO II X68000 新Jコース

CZ-604C TN(本体)……¥348,000
CZ-606TD(0.31カラーディスプレー)……¥79,800
CZ-8NJ2(インテリジェントスティック)……¥23,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥474,600 ➡ 現金大特価

¥6,700×78回	ボなし 頭なし
¥8,000×60回	ボなし 頭なし
¥12,000×36回	ボなし 頭なし
¥17,300×24回	ボなし 頭なし

X68000PRO II X68000 新Eコース

CZ-653C(本体)……¥285,000
CZ-6050(0.39ステレオスピーカ付テレビ)……¥115,000
上海II……¥6,800
倉庫番パーフェクト……¥6,800
TETRIS……¥6,800
信長の野望戦国群雄伝……¥9,800
スーパー大戦略68K……¥8,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥457,000 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000PRO II X68000 新Pコース

CZ-652CBK(本体)……¥298,000
CZ-6020 BK(0.39カラーディスプレーテレビ)……¥99,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト(人気ソフト上記よりお選び下さい)……¥サービス

定価合計 ¥415,800 ➡ ¥257,000

¥6,700×55回	ボ¥20,000 頭なし
¥8,000×60回	ボなし 頭なし
¥12,000×24回	ボ¥20,000 頭なし
¥17,300×18回	ボ¥30,000 頭なし

SHARP

X68000お買上げのお客様へ
各コースで御希望ソフトは「サンダーブレード」「ダウンタウン熱血物語」「ニュージーランドストーリー」「沙羅曼蛇」「ツインビー」「フルスロットル」「パックマニア」「ビーチバレー」「アルカノイド」「熱血高校ドッヂボール」のうちいずれかからお選び下さい

X68000EXPERT II X68000 新Lコース

CZ-653C(本体)……¥285,000
CZ-6050……¥115,000
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥418,000 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

全国出張サポート★
私共にてご購入いただいたX68000は全国出張サポートがうけられます。

X68000PRO II X68000 Fコース

CZ-653C(本体)……¥285,000
CZ-6020(0.39カラーディスプレーテレビ)……¥99,800
住友3M 5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥402,800 ➡ 現金特価

¥7,200×48回	ボなし 頭なし
¥13,100×24回	ボなし 頭なし

X68000PRO II X68000 Gコース

CZ-653C(本体)……¥285,000
CZ-606D(0.31カラーディスプレー)……¥79,800
住友3M 5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥387,800 ➡ ¥249,800

¥6,900×48回	ボなし 頭なし
¥12,600×24回	ボなし 頭なし

X68000 EXPERT II

X68000EXPERT II X68000 新Kコース

CZ-653C(本体)……¥285,000
CZ-604D(0.31ステレオスピーカ付ディスプレー)……¥94,800
御希望ゲームソフト……サービス

合計 ¥379,800 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000EXPERT II X68000 新Cコース

CZ-603C(本体)……¥338,000
CZ-6050(0.39ステレオスピーカ付テレビ)……¥115,000
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥475,000 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000EXPERT II X68000 Aコース

CZ-603C(本体)……¥338,000
CZ-606D(0.31カラーディスプレー)……¥79,800
住友3M 5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥440,800 ➡ ¥305,000

¥8,300×48回	ボなし 頭なし
¥15,200×24回	ボなし 頭なし

X68000EXPERT II X68000 新Rコース

CZ-603C(本体)……¥338,000
CZ-6050(0.39ステレオスピーカ付テレビ)……¥115,000
リップスティックアドベンチャー……¥6,800
ヴァリスII……¥9,800
V'BALL……¥7,900
パワフル麻雀2……¥7,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥503,300 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000EXPERT II X68000 新Dコース

CZ-603C(本体)……¥338,000
CZ-604D(0.31ステレオスピーカ付ディスプレー)……¥94,800
住友3M5 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥450,800 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

SUPER HD

X-68000 SUPER HD X68000 Oコース

CZ-623CTN(本体80M HDD内蔵)……¥498,000
CZ-606DTN(0.31カラーディスプレーチタン)……¥79,800
住友3M 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥600,800 ➡ 現金大特価
安すぎて表示できません
クレジットでもお申し込み出来ます

SHARP

X68000 SUPER HD X68000 Iコース

CZ-623CTN(本体80MHDD内蔵)……¥498,000
CZ-602DBK(0.39カラーディスプレーテレビ)……¥99,800
住友3M 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥615,800 ➡ ¥450,000

¥10,300×60回	ボなし 頭なし
¥15,500×36回	ボなし 頭なし

買ったお客様でしかわからないこのサービス! ——ぜったいX68000を買うならアオヤマがオトク
● 以前当社にてX68000及びX-1を御購入いただいたお客様に限り、CZ-8PC5〔定94,800〕を大特価にてお届けいたします。会員の方は会員ダイアルにてCall!
● X68000をセットでお買い上げいただいたお客様に限り、アスキーターボステックを特価¥2,980、XE-1PROを特価¥6,700また CTRACEを特価¥39,800にて CZ-8NJ2(インテリジェントコントローラ)〔¥23,800〕を特価¥16,900にてお届けいたします 御注文の際に合わせてお申し込み下さい

X68000をはじめソフト&周辺機器類は、当社池袋店・札幌店・旭川店・福岡店にて実演中です 各店X68000コーナーが常設されております

X68000ソフト&周辺機器

SCSIボード(CZ-6BSI)	¥29,800➡現金特価	BF-68PRO	¥19,800➡¥16,800	Communication PRO-68K	¥19,800➡現金特価
システムサコムMIDIボード(SX-68M)	¥19,800➡¥15,300	CZ-6TU	¥33,100➡現金特価	Stationary PRO-68K	¥14,800➡現金特価
LANボード	¥268,000➡¥201,000	オムロンMD-24FP4 II	[illegible]➡現金特価	DATA PRO-68K	¥58,000➡¥43,500
RS-232Cケーブル(平行)	¥7,200➡現金特価	オムロンMD-24FP5 II	¥42,800➡¥33,000	BUSINESS PRO-68K	¥68,000➡¥51,000
RS-232Cケーブル(クロス)	¥7,200➡現金特価	ローランドMT-32	¥64,000➡¥54,400	NEW Printshop PRO-68K	¥19,800➡現金特価
インテリジェントコントローラ	¥23,800➡¥18,900	Hyperword	¥39,800➡現金特価	グラフィックライブラリ vol.1	¥8,800➡現金特価
トラックボール	¥13,800➡¥12,000	CYBERNOTE PRO68K	¥19,800➡現金特価	グラフィックライブラリ vol.2	¥8,800➡現金特価
ジョイカード(延長コード付)	¥3,200➡¥2,900	C compiler PRO-68K	¥44,800➡¥33,600	Musicstudio PRO-68K ver1.1	¥28,800➡¥21,600
CZ-8BS1(X-1用)	¥23,800➡現金特価	CARD PRO-68K	[illegible]9,800➡現金特価	MUSIC PRO-68K (MIDI)	¥28,800➡現金特価
拡張I/Oボックス	¥88,000➡現金特価	CARD PRO システム手帳リフィル集	¥9,800➡現金特価	ソングライブラリ 101曲集	¥8,800➡現金特価
アンプ内蔵スピーカーシステム	¥36,600➡現金特価	CARD PRO 活用フォーム集	¥9,800➡現金特価	Sampling PRO-68K	¥17,800➡現金特価
システムラック	¥44,800➡¥35,800	SX-WINDOW ver1.0	¥6,800➡現金特価	SOUND PRO-68K	¥15,800➡現金特価

X68000シリーズ周辺機器

CZ-8NS1	¥188,000➡¥141,000	CZ-8PC5	¥94,800➡現金特価	I/Oデータ 2MB増設RAM	¥50,000➡¥36,500
CZ-6BN1	¥29,800➡現金特価	IO-735X	¥248,000➡現金特価	I/Oデータ 4MB増設RAM	¥88,000➡¥64,000
CZ-6VT1	¥69,800➡¥52,400	CZ-8PK10	¥97,800➡現金特価	GP-IBボード	¥59,800➡現金特価
CZ-6BV1	¥21,000➡現金特価	1MB増設RAM(CZ-600C専用)	¥35,000➡¥28,000	増設用RS-232Cボード	¥49,800➡現金特価
CZ-6PV1	¥198,000➡¥148,500	1MB増設RAM	¥28,000➡¥22,400	ユニバーサルI/Oボード	¥39,800➡現金特価
CZ-8PC3	¥65,800➡¥39,000	2MB増設RAM	¥79,800➡現金特価	数値演算プロセッサ	¥79,800➡現金特価
CZ-8PG1	¥130,000➡¥97,500	4MB増設RAM	¥138,000➡現金特価	FAXボード	¥79,800➡¥55,800
CZ-8PG2	¥160,000➡現金特価	I/Oデータ 1MB増設RAM	¥25,000➡¥18,000	MIDIボード	¥26,800➡現金特価

X68000万全のサポート AOYAMAにて購入のX68000は万一故障の場合でも全国どこでも出張サービスがうかがいます。万一の場合ワールドインアオヤマサポート係にお電話下さい。お客様のお名前と電話番号だけで手続きは完了。

組合せ自由	激安金利にキャンパスクレジット	ゆっくり、お支払いは8ヵ月先から
各コース以外の組合せもコースをベースに周辺を合せたセットお支払いだって御希望のパターンをお組みいたします さあ、ご相談もお見積りも受注センターもしくは各店へお気軽に	手続きカンタン、大学生の為の超低金利クレジット 20歳以上の学生の方は原則として保証人様には連絡いたしません	クレジット業界最低の金利を有効に使って、支払いは最長8ヵ月後から始まるクレジットでも。

SHARP X68000

パーツセット

最新お買得ソフトセット X68000 パーツセット1

シム・シティー……¥9,800
ナイトアームズ……¥9,700
サイバリオン……¥8,800
グラナダ……¥8,800
きゃんきゃんバニー……¥6,800
スターシップランデブー……¥8,800

定価合計 ¥51,700 ➡ ¥39,900
安すぎて表示できません
クレジットでもお申し込み出来ます

ステレオMIDI音源セット X68000 パーツセット2

CZ-6BM1(MIDIボード)……¥26,800
CM-32L(ローランドMIDI音源)……¥69,000
MA-12AV×2(ホーンアンプ内蔵スピーカー)……¥28,000
MUSIC-PRO-MIDI(ソフト音源)……¥28,800
ソングライブラリ(ソフト音源データ)……¥8,800

定価合計 ¥161,400 ➡ ¥134,900

¥6,900×24回	ボなし 頭なし
¥12,800×12回	ボなし 頭なし

ステレオMIDI音源セット・オールブラック X68000 パーツセット3

SX-68M(MIDIボード)……¥19,800
MT-32(ローランドMIDI音源)……¥64,000
MA-12AV×2(アンプ内蔵スピーカー)……¥28,000
CF-10(コンピュータミキサー)……¥22,000
PC-200(キーボード)……¥36,000
MU-1(ソフト音源)……¥19,800
ソングライブラリ(ソフト音源データ)……¥8,800
—お買上げのX68000と合せて御利用いただけます。—

定価合計 ¥198,400 ➡ ¥163,000

¥8,300×24回	ボなし 頭なし
¥15,300×12回	ボなし 頭なし

カラーグラフィックセット X68000 パーツセット4

GT-1000(カラーハンディスキャナ)……¥79,800
=5220(接続ケーブル)……¥7,500
Z's STAFF・PRO(グラフィック作成ソフト)……¥58,000

定価合計 ¥145,300 ➡ ¥103,800

¥5,300×24回	ボなし 頭なし
¥9,800×12回	ボなし 頭なし

MUSIC PRO-68K ¥18,800➡現金特価, OS-9 X68000 ¥29,800➡¥22,300 Human68K ver2.0 ¥9,800➡現金特価 たーみのるII ¥12,800➡現金特価 Z'S STAFF PRO-68K ¥19,800➡現金特価 C-TRACE ¥68,000➡¥38,000 Kamikaze ¥68,000➡¥49,600 Canvas PRO 68K(ドローグラフィックツール) ¥29,800➡現金特価

一歩ふみこんだアートの世界 X68000

CZ-604C(本体)
CZ-602D(0.39カラーディスプレー付ディスプレー)
住友3M2HD ブランクディスケット
御希望ゲームソフト

定価合計 ¥465,000 ➡ 現金大特価

¥2,800×60回	ボ¥30,00
¥5,000×36回	ボ¥40,00
¥7,800×60回	ボなし
¥11,900×24回	ボ¥30,00

高解像度カラーで金や銀まで表現 X68000

CZ-604C(本体)
CZ-605D(0.39ステレオディスプレーテレビ)
住友3M2HDブランクディスケット
御希望ゲームソフト

定価合計 ¥481,000 ➡ 現金大特価

¥4,900×60回	ボ¥20,00
¥7,300×36回	ボ¥30,00
¥8,900×60回	ボなし
¥17,800×24回	ボなし

グレー限定お買得セット X68000

CZ-603CGY(本体)
CZ-603DGY(カラーディスプレー)
ステレオスピーカー
住友3M2HDブランクディスケット
御希望ゲームソフト

定価合計 ¥450,800 ➡ ¥29[illegible]

¥6,100×72回	ボなし
¥6,900×60回	ボなし
¥10,300×36回	ボなし
¥14,900×24回	ボなし

グレー限定お買得セット X68000

CZ-652CGY(本体)
CZ-604DGY(0.31ステレオスピーカー付ディスプレー)
住友3M2HDブランクディスケット
御希望ゲームソフト

定価合計 ¥410,800 ➡ 現金大特価

¥5,400×72回	ボなし
¥6,100×60回	ボなし
¥9,100×36回	ボなし
¥13,100×24回	ボなし

チタンカラーも入ってますます充実 X68000

CZ-604CTN(本体)
CZ-606DTN(0.31カラーディスプレー)
住友3M 5 2HDブランクディスケット
御希望ゲームソフト

定価合計 ¥400,800 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

AOYAMAだからオトク パソコン通信セット〔ソフトたーみのる2(¥12,800)+モデムMD24FP5II 2400ボモデム(¥42,800)〕➡¥42,800 通信セット〔ソフトX Talk-68K(¥12,800)+モデムMD12FS1200ボモデム(¥21,000)〕➡¥27,300 X68000接続電子手帳セット〔ケーブルCE-300L(¥3,500)+サイバーノート68K(¥19,800)+電子手帳PA-8500(¥28,000)〕➡¥38,600 ★CU-21HD〔ステレオスピーカ付21インチディスプレー〕…¥148,000➡現金大特価 ★CZ-604D〔ステレオスピーカ付603Dディスプレー〕……¥93,000➡現金大特価

ワールドインアオヤマの9ハートサービス

実績のメンテナンス/完全バックアップシステム

● パソコンのセオリーが体験できる。● パソコンだけのスペース。

池袋本店OPEN!
1/17 (TUE) 12:00

Computer PLAZA
CREATIVE LIFE STORE TOKYU HANDS
東急ハンズ池袋店となり
AMLUX 前

ワールドインアオヤマにおまかせ下さい。

— クレジット —
業界一番のスーパークレジットで。
学生の味方、キャンパスクレジットがますますワイドに。
お支払いはナント!84回まで。
ゆうゆうお支払いは8ヶ月先から。
— サポート —
万一のときも完全バックアップ。
— 会員・保険 —
最先端の"CLUB246"。
インシュアランス・サービス
グーンとお得な下取りシステム。
— ショップ —
全店統一のサービス 見て、触れて、納得。
BEST SHOPS BEST SERVISE。
— サービス&ポリシー —
グーンとお得な下取りシステム。

返品・交換について

返品・交換は商品到着後3日以内にお願いいたします。

〈返品・交換をお受けする場合〉
●不良品(キズ・汚損)が届いた場合
●ご注文商品と違う商品が届いた場合
上記の場合、返送料は当社が負担いたします。ただしお客様の事由(注文ミスなどによる返品の場合、返送料はお客様の負担となりますのでご了承下さい。
〈次の場合返品・交換はお受けできませんのでご注意下さい。〉
1 1度ご使用になった商品 2 お客様の責任でキズや汚れが生じた商品
3 全ソフト商品

日中ご不在がちな方はお振込

昼間ご不在がちの方、お急ぎの方は銀行振込が便利です。
お申し込みからお届けまで、お時間をとりません。
● 銀行振込みの場合

取引銀行	住友銀行 池袋支店
口座番号	普通 1065392
口座名	株式会社 ワールドイン アオヤマ

返品・交換についてのお問い合わせは
お客様相談室 03-987-7795

電話でのご注文の場合

平成3年1月1日から東京は局番が4桁になり頭に3がつきます

03-3987-7771

お電話番号はおかけ間違いのないようにお願いします。
北海道受注センター 011-251-6771
九州受注センター 092-672-7771
お好きな時間にお電話を!

テレフォンショッピング

電話受付時間 ■月曜日〜金曜日 10:30〜20:30 ■土・日曜日・祭日 10:30〜19:00

クレジットカードもOK!

クレジットカードをお持ちの方お支払は1回払いです。お申し込みの際、①カード名 ②会員No. ③有効期限をご連絡下さい。
カードでお申し込みの場合、販売価格が変わりますのでお電話にてお問い合せ下さい

サポート

〈万一のときにも完全バックアップ〉
万一の初期不良があった場合でも当社では万全の体制でお客様をフォロー致します

クレジットご利用のご[illegible]

手続きはお電話でかん[illegible]
お電話をいただきましたら、支[illegible]き落とし口座とお客様が不安に[illegible]いたことがすべて解消 専門の[illegible]トスタッフがお役に立ちます

業界一番のスーパークレ[illegible]
ご都合に合わせて自由にお支[illegible]金額を設定できます
手続きはとても簡単、そしてス[illegible]もちろん手数料は超低金利
お支払いも月々1,000円から[illegible]お得に使いやすくなったアオヤ[illegible]ジットです

学生の味方キャンパスクレジ[illegible]ますま[illegible]
キャンパスクレジットのワクが[illegible]がりました 最高10万円までの[illegible]でき、手続きもその場で即OK[illegible]も月々1,000円からと、とっても[illegible]アオヤマは皆さんを応援します

お支払いはナント84回[illegible]
お支払いは少なく、ゆっくりと ワール[illegible]オヤマなら最長84回まで自由にセ[illegible]ボーナス額を増やしたり減らしたり[illegible]ん、ボーナス一括や二括払いもOK

ゆうゆうお支払いは8ヶ月[illegible]
欲しい機種は今すぐ欲しい!
だけど以前のローンが残ってい[illegible]就職してから始めたい
という方にお勧め
8ヶ月先までならいつ始めても[illegible]これならとっても安心ですね

● 配送日指定OK ● 土・日配送OK ● 札幌PAXに買取専門ショップOPEN ● 専門スタッフが常時待機 ● 中古パソコン4ヶ月間保証 ● FAX見積りサービス ● もちろんクレジットカードも使えます ● カード払いでキャッシュレス

お客様相談室 03-3987-7795 すでにご注文いただいているお届け時間(時期)やメンテナンス、その他のお問い合わせは右記へお電話下さい。

旭川店
0166-25-7771
旭川市4条8丁目ツジビル
■営業時間 11:00〜19:00

札幌店
011-251-1777
札幌市中央区南2条西3丁目
リンクエギビル3F
■営業時間 11:00〜19:30

札幌 PAX 店 WORLD IN AOYAMA
011-251-5315
札幌市中央区南2条西2丁目
ブロックビル6F
■営業時間 11:00〜19:30

池袋店ソフト店
03-3989-1171
豊島区東池袋1-28-6
パールシティビル2F
■営業時間 11:00〜19:00

池袋ショールーム
03-3989-7771
豊島区東池袋1-27-12明治生命池袋ビル9F・首都高速道路側
■営業時間 11:00〜19:00

福岡店
092-716-77[illegible]
福岡市中央区渡辺通り4-9-25
ユーテックプラザ4F・地下鉄天神駅[illegible]
■営業時間 11:00〜19:30

平成3年1月1日より東京の局番が4桁になり頭に3がつきました。

Dungeon Master

CHAOS STRIKES BACK

勇者たちよ!今一度立ち上がれ。

勇者達の戦いが再び始まった。

「ロード・カオス」を倒し世界に平和と秩序をもたらした勇者達、・・・しかし「ロード・カオス」は生きていた 秘かに新しいダンジョンを作り悪の力を蓄えていたのだ

発売以来再び巻起こる興奮!!

リアルタイムRPGの最高峰「ダンジョンマスター」の続編は、アニメを使ったイントロ、FM音源対応の音と音楽、好きな顔に書き換えられるキャラクターエディット機能、新しい魔法や機能の追加など前作以上にパワーアップ

続 ダンジョン・マスター カオスの逆襲

熱狂的に発売中! X68000版

■FM-TOWNS版■PC-9801VM21以降 UV21以降 各¥9,800(税抜)

(注)PC-98DO及び、PC-9801VM21/VM11/CV/UV21/UV11でもゲームはできますが、動作スピードが遅くなります

© 1990 SOFTWARE HEAVEN, INC./FTL GAMES. LICENSED THROUGH AN AFFILIATION WITH J.P. INTERNATIONAL. © 1990 VICTOR MUSICAL INDUSTRIES, INC.

スーパー・ゲームミュージック・セレクション

© 1990 FTL GAMES/SOFTWARE HEAVEN, INC.

「ダンジョン・マスター」全世界を熱狂させた「ダンジョン・マスター」イメージアルバム!

この1枚でダンジョン・マスターの世界が広がる!「ダンジョン・マスター」から「カオスの逆襲」まで1枚に収録

2月21日発売予定

シューティングの極み!!

ゲーム性、グラフィックス、サウンド何もかもがX68000の限界を超えた!!

■驚異の迫力で展開する変化に富んだ全10ステージ。■各ステージ毎に武器(5種類)の選択/ステージの解説の表示をし、その間のプログラムのロードにより途中のディスクアクセスがなく、スムーズにプレイできます。■大型エネルギー・ゲージ/スコアー・ゲージの採用による迫力ある戦闘がたのしめます。■戦闘中のスピード変更機能搭載による状況に応じた臨機応変の戦いを実現。■快感のテーマ曲ほか全20曲収録。内蔵音源に加えてMIDI音源にも対応。迫力のサンプリング効果音も搭載。

好評発売中!

本格的3D快感シューティングゲーム

●X68000対応 ¥8,800(税抜)

ニューラル・ギア

企画・開発:Fill in Cafe

発売:ビクター音楽産業株式会社

通信販売 当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業(株)〈通信販売係〉

SHARP X68000専用ハードディスク

HXD040(1台目用外付モデル)

Xstor40はシャープX68000専用に開発されたハードディスクです。目的に応じて外付タイプ(2モデル)と内蔵タイプ(1モデル)を用意。従来の汎用サブシステムにはない数々の特徴とハイセンスなデザインを実現した省スペースタイプの高品質なハードディスクです。

■平均アクセスタイム23ms。又バッファサイズ、32Kバイトを装備。満足のいく高速性能を提供。
■パーソナルには余裕の40Mバイトの記憶容量。更に増設用HXD042を付加することにより最大80Mバイトまでのディスクシステムが利用可能。
■Human 68K(Ver1.00以上)、OS9対応。既存の多くのソフトウェアがそのまま利用可能。
■交替セクタをユーザー領域から独立。しかもFormatプログラムにより自動実行。
■切電時のオートパーキングロックを採用。不意な衝撃に対しても磁気面を保護。
■高品質、低価格を実現。

HXD040:Xstor40/1台目用外付モデル……………¥118,000
(X68000/ACE/EXPERT/PRO用)
HXD042:Xstor40/2台目増設用外付モデル………¥128,000
(X68000/ACE〈HD〉/EXPERT〈HD〉/PRO〈HD〉 HXD040又はHXD140の増設用)
HXD140:Xstor40/内蔵用モデル……………………¥98,000

●テータ転送速度/1.5MB/S ●インターフェース/SCSI(シングルユーザ) ●交替処理/FORMATコマンドによるセクタ単位の自動交替処理 ●外形寸法/35H×155W×313Dmm(HXD040/HXD042)/135H×155W×41Dmm(HXD140) ●重量/約2.5kg(HXD040/HXD042)/約800g(HXD140)

詳しいカタログが必要な方は本社までご請求下さい。
※内蔵用モデルの対応機種については、お問合せ下さい。

HXD140(内型用モデル)

株式会社アイテム
本社/〒251 神奈川県藤沢市南藤沢8-1-202
TEL.0466-27-1668代 FAX.0466-27-2800
東京ショールーム/〒105 東京都港区新橋4-31-7中村ビル7F
TEL.03-3434-4171 FAX.03-5472-5315

グラディウスに続け！　今月もX1turbo用同人ソフトだ。紹介するのはMoon Light Magicによる電脳絵本といった感じのSFアドベンチャーゲーム。5″2D2枚組のボリュームでお届けする。

●

先月はお休みしたが，12月号で紹介したX1turbo版グラディウスの反響はなかなか凄かった（当然か？）。

その後，編集室に現れた横内君，初めて見たというX68000版グラディウスではとまどいながらもいきなり2周目の最終面に突入してしまうあたりはさすがといえる。横内君もいまは受験生。今後の展開に期待したい。

さて，今回紹介するのは「Moon Light Magic」によるX1turbo用（2ドライブ要）のSFアドベンチャーゲームだ。

宇宙船の事故に巻き込まれた主人公フェリサと謎の男ガイの物語。唐突に事故直後から始まり一気にラストシーンまで突っ走る。シナリオはストーリーに依存するため基本的に一本道の構造となっている。コマンド選択式ということもあり，ふつうにやっていけば絶対にいきづまることはないだろう。決して単調な展開ではない。かなり大胆に話が進んでいくのだが，途中の枝葉が少ないので展開がさらに早く感じる。

全体にコマンドに対するキャラクターのレスポンスがいい。変にSFとかストーリー重視というのでなく娯楽作品に徹しているのが成功しているといえるだろう。

グラフィックは見てのとおり，枚数もかなり多い部類に入るだろう。音楽も気合いが入っている。オリジナル曲13曲がOPM版とPSG版で用意されており，どれもかなりがんばっている。ミュージックモードもあるぞ。

特に後半はカット割りが細かい。その際，グラフィック表示のためのディスクアクセスが若干長めに思える（5秒弱）。しかし，ディスクアクセス中でもちゃんと音楽演奏していることを思えば，かなり健闘しているといえるだろう。X1turbo専用と銘打つだけはあるということか。

全体に作り慣れた感じで安定感がある作品だ。

入手方法

価格は送料込み1,200円。購入希望の方は例によって，

1) 申し込み内容を書いた手紙
2) 料金分の無記名の郵便小為替
3) 返送先を書いた宛名シール

を用意し，

〒273-01
千葉県鎌ヶ谷市鎌ヶ谷1-12-37 稲田和明

まで封書で問い合わせてほしい。

なお，売買に関する問い合わせやトラブルに，編集室では対応できないので注意してほしい。

予告

X1特集というわけではないのだが，来月もX1関係のゲームを紹介する予定。キーワードは「X1turbo Z専用」だ。

1990年度 GAME OF THE YEAR

ノミネート作品発表

1990年度のゲームソフトの傾向と対策

さあ。さあさあ。来ましたよ，これが。GAME OF THE YEARのノミネート発表が。見てください，ずらり揃った顔ぶれを。どれも，こいつを抜きにしちゃ今年のゲーム界を語れないぜ，という強豪ばかり。確かにどれも人気があるし，優劣なんてつけられないかもしれない。しかし，'90年のゲームをきっちり締め括ってこそ，これからのゲームが楽しみになろうというもの。ここはあえて「オレはこいつを推すぜっ」という読者の皆さんの声で'90年のベストゲームを決め，頭をなでてやろうじゃないですか。

で，このGAME OF THE YEARですが，今年から大きく変わりました。各賞が整理されてご覧のとおり。あやや，ちょっと寂しいんじゃないの，という気もしますが，みんなで決めるものはキッチリ決めて，あとは独断と偏見たっぷりの「勝手にGAME OF THE YEAR」で盛り上がろうというコンセプトなのです。今年はスタッフの乱入も予想されますので，それに負けない気合いの入ったハガキをお待ちしております，ハイ。

そしてノミネートの選考方法。Oh!Xゲーム大賞については後出のチャートの上位から選び出し（1989年度以前の作品は除外），さらに1990年12月号掲載の作品の中から上位に入る実力があると思われるものを考慮しました。ほかのノミネートに関しては従来どおりの推薦です。しかしそこはそれ，GAME OF THE YEARだからして乱入投票もなんでもアリ。異議のある方は遠慮なくかかってらっしゃい。

さらに，今年から皆さんにとってのゲームライフがどうだったか，'90年を振り返ってのハガキを募集したいと思います。「春の新作，あれとこれで迷ったけどこっちにしちまったぜ！」とか「これを買ったとたんにこのソフトが発売決定してくやしかったっ！」という声を全国に轟かせてやるという方，ぜひふるってご応募ください。のどから手を出してお待ちしております。おえ。

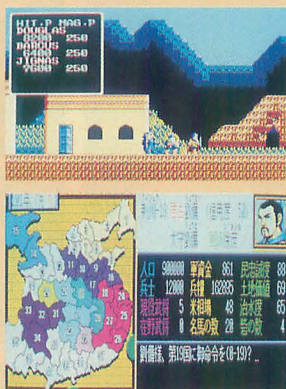
さあ能書きはここまでだ。とにもかくにも君のハガキがなけりゃ始まらない。GAME OF THE YEARの栄冠を手にするのは誰か!?

決めるのは，君だっ!!

選択応募部門

Oh!Xゲーム大賞

GAME OF THE YEARの大黒柱ともいえるのがこのOh!Xゲーム大賞です。'90年のゲームシーンをもっとも賑わし，ゲーマー達の心を熱くさせたゲームに贈られる，よーするに「とにかくあんたがいちばん偉かった！」賞なわけ。この賞を取った強者こそ，'90年を象徴するにふさわしいゲームなのです。

しかし，今年は「これが強かった」と単純には括れないほどゲームの多様化・複雑化が進んだのもまた事実。ノミネート作品もそれを表すように多種多様な性格のソフトが揃っています。どれも「今年の思い出のゲーム」として支持する人がいそうなだけに，どれが大賞を取るかはひとえに熱心なファンの多さにかかっているという感じもしますね。

- ダンジョン・マスター(X68000)
- ポピュラス(X68000)
- ソーサリアン（含シナリオ集）(X1)
- ワンダラーズ・フロム・イース(X68000)
- スーパーハングオン(X68000)
- シムシティー(X68000)
- ラグーン(X68000)
- グラナダ(X68000)
- 三国志II(X1/X68000)
- メタルサイト(X68000)
- アルガーナ(X1)
- バブルボブル(X68000)
- 天下統一(X68000)
- ナイトアームズ(X68000)
- ナイアス(X68000)

グラフィック賞

X68000はずばぬけたグラフィックパワーがあるだけに，ソフトハウス間のグラフィックに関する力量の差が表れやすいということができます。そんな厳しい環境のなかで，グラフィックに関して優れた美的センスと演出のテクニックを発揮していたと誰もが認めることのできるものがこれら6作品です。どれも単に綺麗に描かれているというだけでなく，キャラ同士の対決や，シナリオと連動した演出など，画面をうまく使うということに関しては，ほかにはないなにかを持っているものばかりです。

- 機甲師団(X68000)
- グラナダ(X68000)
- ナイトアームズ(X68000)
- 闇の血族（前・完結編）(X68000)
- ラグーン(X68000)
- ワンダラーズ・フロム・イース(X68000)

音楽賞

ゲームを盛り上げる手段としてすっかり定着した音楽ですが，今年の大きな特徴としてはMIDI対応が進んだことが挙げられるでしょう。特にモトスのように3種類の音源に対応するほど力を入れた作品も登場し，MIDIボードはゲーマーの間でも必需品となりつつあります。

また，単にBGMとしてだけではなく，シナリオ・映像とシンクロした音響効果として使いこなした作品が出てきたことも今年の特徴のひとつで，そういった作品への評価がどうなるか注目されます。

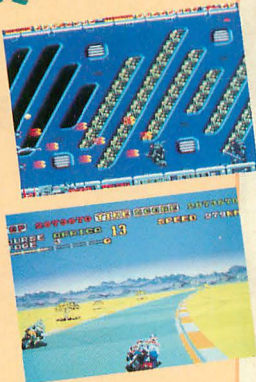
- グラナダ(X68000)
- ナイトアームズ(X68000)
- メタルサイト(X68000)
- モトス(X68000)
- 闇の血族（前・完結編）(X68000)
- ラグーン(X68000)

プログラミング技術賞

以前の「特殊演出部門賞」に代わって設置されたのがこの賞です。単にプレイヤーへのアピール度が高いということのみならず，高い技術力でマシンの限界を引っぱり上げたゲームに対して贈られます。

ノミネート作品を見ると，画面関係のテクニックに目がいきがちという気もしますが，それを差し引いてもラスタースクロールなどのテクニックがX68000のゲームに与えた影響は大きいものがありました。

また，アルガーナの5重スクロールはまさに圧巻。これが究極というX1のテクニックを見せてくれたことも見逃せません。

- アルガーナ(X1)
- グラナダ(X68000)
- スーパーハングオン(X68000)
- ナイアス(X68000)
- ナイトアームズ(X68000)
- メタルサイト(X68000)

ゲームデザイン賞

今年のゲーム界が受けたもっとも大きな衝撃が，このゲームデザイン面でしょう。いうまでもなく海外ソフトの登場によるものです。

海外の3作品はいずれもジャンル分けが通用しないほどのオリジナリティを持っており，似たもの同士の多い日本のゲーム界ではいっそう熱烈なゲーマーの支持をもって迎えられました。

一方，日本でもリアルタイムのシミュレーションという独自のスタイルを構築した機甲師団など，大技とはいかないまでも光るものを持つゲームが登場しています。

- 機甲師団(X68000)
- シムシティー(X68000)
- 大航海時代(X1)
- ダンジョン・マスター(X68000)
- 天下統一(X68000)
- ポピュラス(X68000)

自由応募部門

主演&助演キャラクター賞

去年はテトリスの直線ブロックとサイバースティックという，フセインもビックリの結果になったこの賞，今年もみんなの思い入れを尊重するということで，あえてノミネートはしませんでした。あなたのこの1年のゲームライフで，忘れられないキャラクターに清き一票を投じてあげてください。

底抜け脱線ゲーム賞

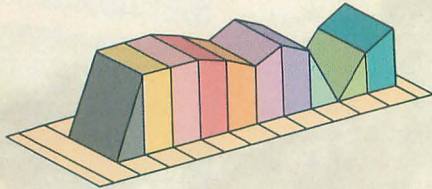
せっかく楽しみにしていたのに，いざやってみると「はぐー」とタメ息のでてしまう，いわゆる「○○ゲー」，そう，あのアイツに対する叫びを誌上で発散させてやろうじゃないかというのがこの賞です。ただし，グチはだめですよ。他人を楽しませることのできるハガキに限ります。また，「みんなは嫌いだろうけど，ぼくはここがいいから好きなんだ」というハガキも受け付けてますよん。

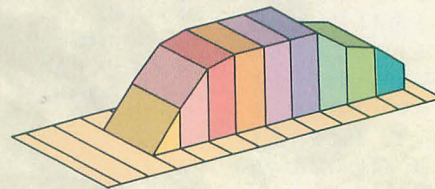
その他自由応募部門賞

あとは読者のみなさんのセンスに任せた自由応募部門賞。もうなんでもアリ。面白ければ通します。ちょっと'90年のことならこれはいわせてくれということがあれば，迷わずこのコーナーにあてて送りましょう。思わず鼻からコーヒーを噴いてしまうようなムチャクチャなハガキを，スペースを空けて待っております。

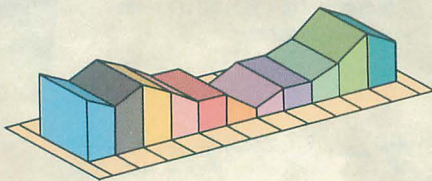
グラフで見るTOP10

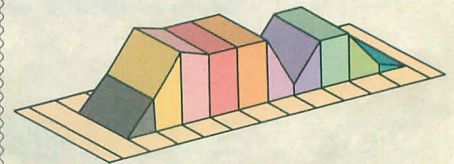
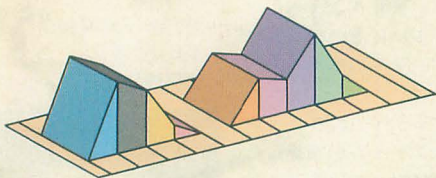
ここでは上位10タイトルのゲームが，この１年間のTOP10の中でどのような動きを見せていたかを，グラフつきで紹介します。コンスタントに人気を得ているもの，また月によっては票が得られなかったものなど，タイプはさまざま。改めて見直すとけっこう面白いもんですね。

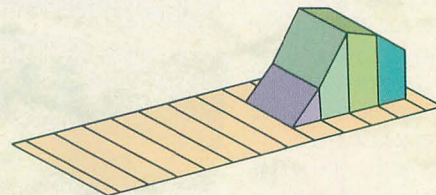
1位 ダンジョン・マスター

登場してすぐに１位をキープ。みんながゲームをクリアする頃にランクを落とし，さらにシムシティーの登場によって票を奪われたが，最近またカオスの逆襲のニュースにつれて持ち直した。

2位 ポピュラス

意外に人気の立ち上がりが鈍かったゲーム。１位になったときはかなりのブッチギリ状態が続いたが，最近になって鎮静化してきた。プロミストランドの発売がもう少しあとだったら……。

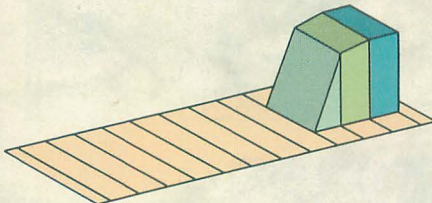
3位 ソーサリアン

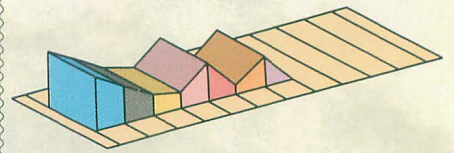
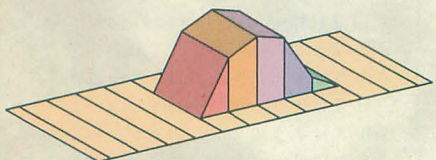
ベテランらしく成績にムラがない。夏場にバテるところまでベテランらしいが，涼しくなると「ここまで来たらあとは１位だ」の声のもとに再びパワーを取り戻した。さすがは大御所といったところか。

4位 ワンダラーズ・フロム・イース

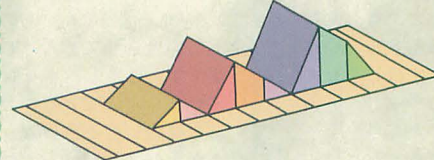
海外ソフトにはさまれてやや不運な境遇だった。が，人気があることには変わりない。また，ラグーン，ギャラガ'88など，アクションものが初登場した月に弱くなるという性質を持っている。

5位 スーパーハングオン

春先，秋など気候がよくなるとふらつくのは，みんな天気がいいと本物のバイクでツーリングに出てしまうからなのか（そんなバカな）？ これぞ先が読めないソフトの典型といえる。

6位 シムシティー

夏の終わり頃から鋭い伸びを見せたこのゲーム。が，ラグーンともつれて足元がおぼつかないという現状を，いまだに引きずっているところがある。今後トップの座に戻ることはできるのだろうか。

7位 ラグーン

シムシティーにひと月遅れて登場。さすが去年の実力者，ズームは得票が安定している。しばらくは現状維持といったところか。今年もチャートを賑わせそうだ。しかし，もう少し発売が早ければ……。

8位 ジェノサイド

アフターバーナー，ファンタジーゾーンより持ちこたえた。ズームファンが「ラグーンが出るまではっ」と粘りながら力つきていくさまがグラフによく出ている。それでも８位は見事なもの。

9位 グラナダ

非常に典型的なスマッシュヒットのグラフパターン。サッと盛り上がってサッと消える「ひと夏の恋」型とでもいえばいいのかな。すぐに忘れて，しかもあとくされなしってヤツですか。

10位 三国志II

最初の波はX1turbo版発売。次の波はX68000版発売決定。最後の波はX68000版発売。光栄のファンは実に正直者揃いだ。しかし，なんで発売されるとすぐさま順位が下がるんだろう？

TOP10 新年早々仕事納め

～'90年のTOP10の動きを斬る!!～

集計したのは'90年1月号から'91年1月号まで（ただし'90年2月号は休み）の12回分。その月の1位のゲームに10点，2位9点……10位1点とポイントを与え，その合計を集計しました。それがこの表です。

No.	ゲームタイトル	ソフトハウス	ジャンル	対応機種	1	3	4	5	6	7	8	9	10	11	12	1月	年間
1	ダンジョン・マスター	ビクター音楽産業	RPG	X68000	0	0	10	10	10	9	7	9	7	0	5	7	74
2	ポピュラス	イマジニア	シミュレーション	X68000	0	0	0	5	9	10	10	10	10	7	6	3	70
3	ソーサリアン	日本ファルコム	RPG	X1turbo	8	6	8	6	4	4	1	3	3	4	7	5	59
4	ワンダラーズ・フロム・イース	日本ファルコム	RPG	X68000	0	0	4	9	8	8	8	1	6	6	1	0	51
5	スーパーハングオン	シャープ/SPS	アクション	X68000	0	9	7	2	0	0	5	4	9	3	0	0	39
6	シムシティー	イマジニア	シミュレーション	X68000	0	0	0	0	0	0	0	0	4	10	9	6	29
6	ラグーン	ズーム	RPG	X68000	0	0	0	0	0	0	0	0	0	9	10	10	29
8	ジェノサイド	ズーム	アクション	X68000	7	5	3	2	5	1	4	0	0	0	0	0	27
9	グラナダ	ウルフ・チーム	アクション	X68000	0	0	0	0	0	7	9	8	2	0	0	0	26
9	三国志II	光栄	シミュレーション	X1turbo/X68000	0	0	0	3	0	6	3	0	8	5	0	0	26
11	メタルサイト	システムサコム	アクション	X68000	3	10	9	0	0	0	0	0	0	0	0	0	22
12	アフターバーナー	電波新聞社	アクション	X68000	10	8	0	0	0	0	0	0	0	0	0	0	18
13	アルガーナ	M.N.M.ソフトウェア	RPG	X1/turbo	0	0	6	0	6	3	0	0	0	0	0	0	15
13	バブルボブル	電波新聞社	アクション	X68000	0	0	0	8	7	0	0	0	0	0	0	0	15
13	天下統一	システムソフト	シミュレーション	X68000	0	0	0	0	0	0	6	7	2	0	0	0	15
16	テトリス	BPS	パズル	X1/turbo/X68000	9	3	0	0	0	0	0	0	0	0	0	0	12
16	ファンタジーゾーン	電波新聞社	アクション	X68000	5	7	0	0	0	0	0	0	0	0	0	0	12
18	ナイトアームズ	アルシスソフトウェア	アクション	X68000	4	1	6	0	0	0	0	0	0	0	0	0	11
18	サイバリオン	シャープ/SPS	アクション	X68000	0	0	0	0	0	0	0	0	0	1	8	2	11
18	パロディウスだ！	コナミ	アクション	X68000	0	0	0	0	0	0	0	0	0	0	3	8	11
21	ギャラガ'88	電波新聞社	アクション	X68000	0	0	0	0	0	0	0	5	5	0	0	0	10
22	ナイアス	エグザクト	アクション	X68000	0	0	0	0	0	0	0	0	0	0	0	9	9
23	サンダーブレード	シャープ/SPS	アクション	X68000	0	0	0	7	1	0	0	0	0	0	0	0	8
23	ワールドコート	SPS	アクション	X68000	0	0	0	0	0	0	0	0	0	8	0	0	8
23	エアーコンバット・遊撃王II	システムソフト	シミュレーション	X68000	0	0	0	0	0	0	0	0	0	0	4	4	8
26	サーク	マイクロキャビン	RPG	X68000	0	0	0	0	0	5	2	0	0	0	0	0	7
27	スタークルーザー	アルシスソフトウェア	RPG	X1turbo/X68000	6	0	0	0	0	0	0	0	0	0	0	0	6
27	ファーストクイーン	呉ソフトウェア工房	シミュレーション	X68000	0	0	1	0	3	2	0	0	0	0	0	0	6
27	A-JAX	コナミ	アクション	X68000	0	0	0	4	2	0	0	0	0	0	0	0	6
27	クォース	コナミ	パズル	X68000	0	0	0	0	0	0	0	6	0	0	0	0	6
31	斬	ウルフ・チーム	シミュレーション	X68000	0	4	0	0	0	0	0	0	0	0	0	0	4
31	夢幻戦士ヴァリスII	日本テレネット	アクション	X68000	0	2	2	0	0	0	0	0	0	0	0	0	4
33	トンネルズ&トロールズ	スタークラフト	RPG	X1/turbo/X68000	0	0	0	0	0	0	0	2	0	1	0	0	3
34	V'BALL	シャープ/SPS	アクション	X68000	2	0	0	0	0	0	0	0	0	0	0	0	2
34	機甲師団	アートディンク	シミュレーション	X68000	0	0	0	0	0	0	0	0	0	0	2	0	2
36	サバッシュ	小学館	RPG	X68000	1	0	0	·0	0	0	0	0	0	0	0	0	1
36	イメージファイト	アイレム	アクション	X68000	0	0	0	0	0	0	0	0	0	0	0	1	1

ではさっそく結果を見てみましょう。注目の年間ランキング1位はァ，

ダンジョン・マスター!!（拍手！）

最大のライバル，ポピュラスをかわしての堂々1位。トップの回数ではポピュラスに負けていますが，その間もしっかり上位をキープし，秋以降はカオスの逆襲のニュースでゲームシーンを賑わせてくれました。斬新なゲームシステムと，そのシステムならではの面白さをキッチリと出している点にはプレイヤーをうならせるものがあります。

また，今年は海外の大作の活躍が目立った年でした。チャートを見ても，ダンジョン・マスター，ポピュラスの強さは群を抜いているし，シムシティーも発売時期の遅さにもめげず6位を獲得しています。

そんな海外ソフトにちょっと押され気味だった国産ゲーム勢。なんと，最高順位をとったのはX1turbo用のソーサリアン。ゲームの完成度とX1ユーザーの執念が揃うと，チャートのここまできちゃうんですねぇ。毎月ランクインしたのは，37本登場したTOP10のソフトの中で，このソーサリアンのみ。拍手拍手。続いてのワンダラーズ・フロム・イースは，期待したほどの支持が集まらなかったようですがそれでも4位。日本ファルコムが国産ゲーム勢の1・2フィニッシュを飾りました。

それを追いかけるのが，X68000専用ソフトと他に類を見ないカリスマ性で支持を集めるズーム。ゲーマーをして「ズームだからよい」といわしめてしまうのは有名な話。ラグーンも7位に入っています。

X68000専用ソフトは去年よりだいぶ数が増え，ランクの上位にも顔を出しています。グラナダ，メタルサイト，ナイトアームズといったところは，どれもX68000ユーザーの優越感をくすぐる秀作でした。順位は低いものの見逃せないのが22位のナイアス。案外ダークホースかも。

さて，ビデオゲームの移植ものではSPSと電波新聞社が双璧ですが，今年はスーパーハングオンを5位に入れたSPSに軍配が上がりました。一方の電波新聞社はややマニアックな要求に応えていたようで，去年発売のアフターバーナーが最高位という結果に終わっています。このジャンルにはコナミ，アイレム，システムサコムなどが続々と参入しており，来年は激戦区になる気配が濃厚。特にコナミのパロディウスだ！は前評判で18位に入るほど。'91年の大きな話題のひとつです。

ランキング全体としては，非常に納得のいく順番になったと思います。集計の方法上，長く遊べるゲームのほうが上位に行きやすいという傾向はありますが，逆に話題性に流されないしっかりしたランキングだともいえるのではないでしょうか。

応募要項

GAME OF THE YEARの投票は，

1. 愛読者カードの記入欄
2. 官製ハガキもしくは封書

のいずれかに応募したい賞の名前とソフト名，そして推薦理由を明記して，Oh!X編集部までお送りください。採用ハガキの中から抽選で2名の方にハンディスピーカー，9名の方にカードゲームをプレゼントします。締め切りは2月15日（消印有効）。たくさんのご応募をお待ちしております。

「1990ゲーム回顧録」のお知らせ

ゲーマーの生活は常に波瀾万丈に満ちている。どのゲームを選び，どれを究めるか。ゲームを巡って日々繰り返されるさまざまな事件・陰謀・策略。時に怒り，時に笑い，また時に涙を流す。ゲーマーは毎日が真剣勝負なのだ。

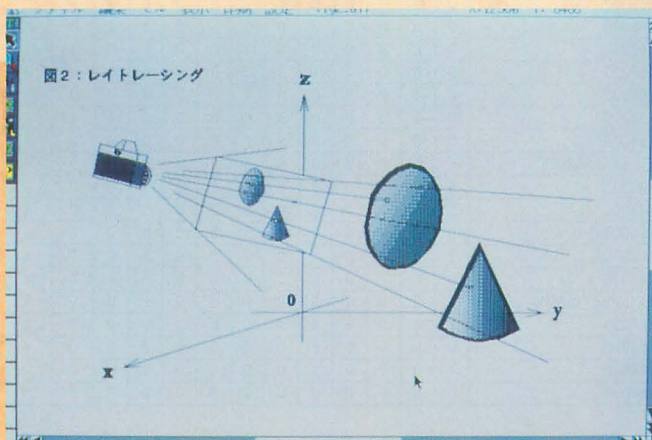
なんてカッコいいもんかどうかは知りませんが，'90年の「ゲームと私」についての意見を募集します。自由応募部門のように単純に「このゲームはこうだった」では括れない話を官製ハガキもしくは封書に書いて送ってください。

いちばん面白かった方にはアイワのカセットボーイをさしあげます。かしこ。

特集

グラフィックの“実験的”手法

CGの始まりはドットの集合体であった。図形を表しうるグラフィック画面を経て，アナログRGBの普及により，我々は連続した色彩の空間を手に入れることとなった。それは従来の「形」を主としたパソコンCGの世界から「質感」までも表現することのできる，そこにひとつの世界を再現できるものだった。
パソコンの計算能力も向上し，その手法や表現力はワークステーションの域にまで近づいた。画面の表現力は時として画材を超えていた。しかし，まだ我々はパソコンCGの可能性のすべてを見ているわけではない。想像の翼は新しい世界を拡大し続けるのだから。

CONTENTS

CANVAS PRO-68Kで使用できるグラフィックデータ集。慣れるまで時間がかかるツールだけに充実したデータ集はかかせない。今後も拡充してほしい。

“実験的”というタイトルがつけられているが，もともとOh!Xのグラフィック特集すべてが実験的といってもいい。

まず，Z'sSTAFFを2Dツールのプラットホームとしてとらえてみた。Z's-EXに拡張された機能には，画像処理，2Dと3Dの融合などテーマは尽きない。

しかし，実験の最たるものは機能拡張そのものだ。それはPICFILER式の呼び出しから始まり，Z'sSTAFFのマスク機能との融合にいたる。現在，多くの種類のソフトでアドイン（Add In）型のツールが出てきている。基本機能はそのままに，必要な機能はあとからユーザーが組み込んで使用することができるのだ。グラフィックの場合，画像処理などのフィルタはアドインソフトにもってこいの存在だ。もともとの強力な機能にあわせ，ユーザーレベルでの拡張の可能性ができたわけだ。

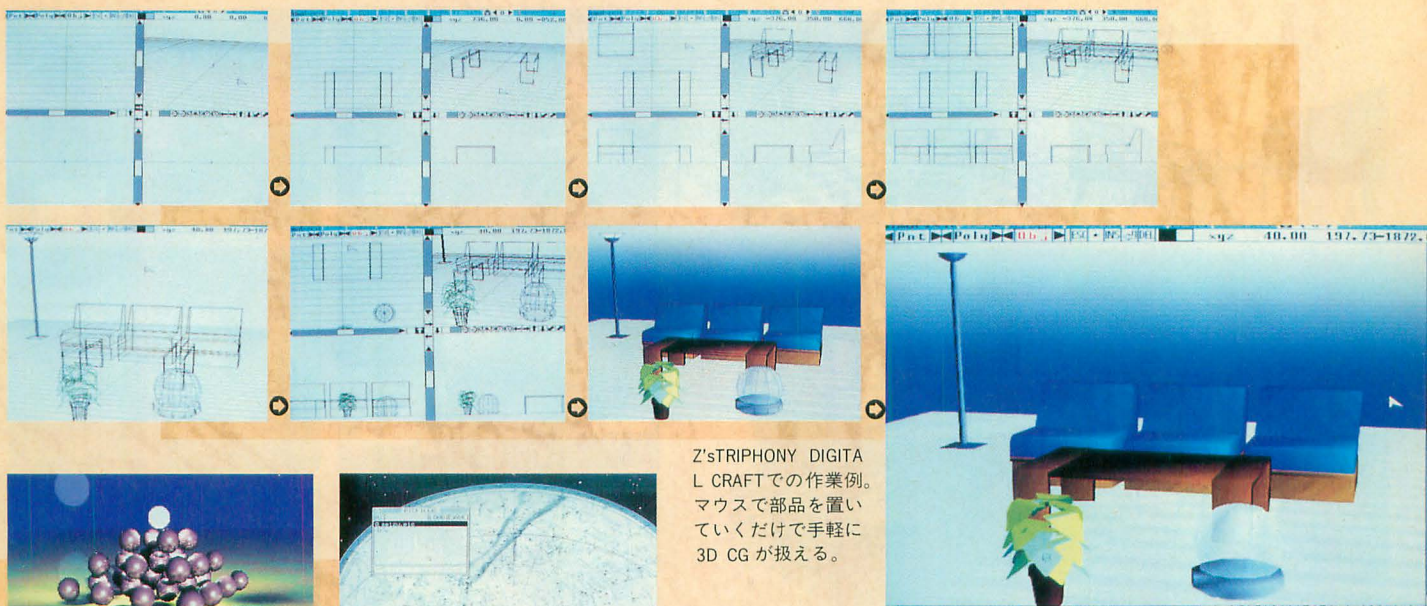
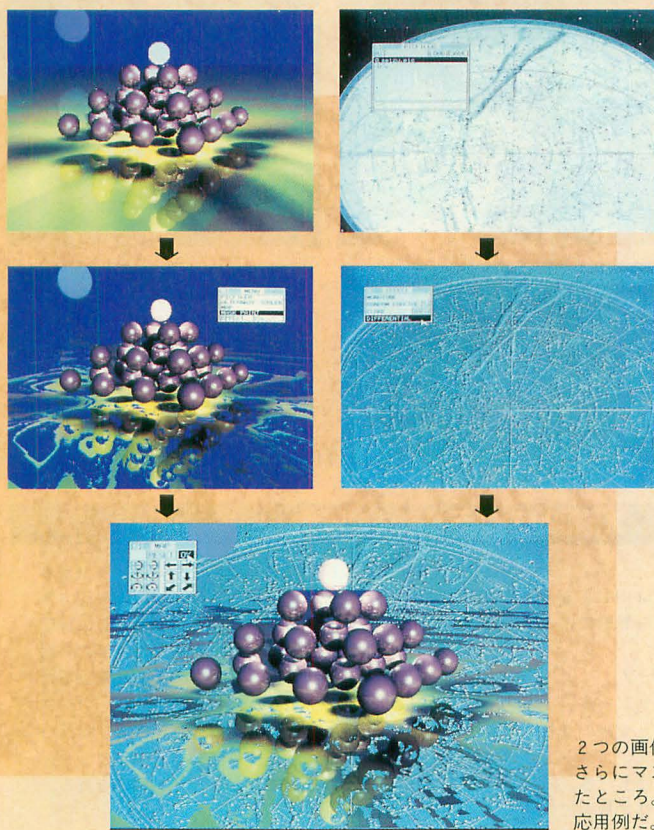
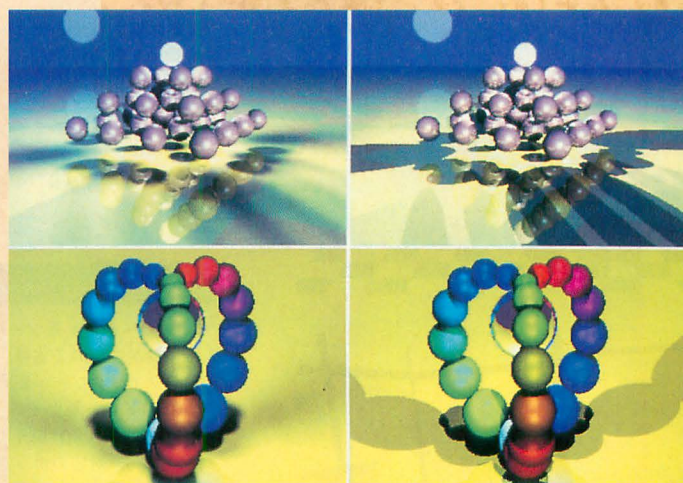
パソコン上のレイトレーシングツールもボクセル分割という高速化の山を乗り越え，ついにポリゴンとメタボールという「自由曲面」への道を歩み始めた。となれば次世代のツールに求められるのは「より自然な画像生成」であろう。

このあいだまでワークステーションがやっていたことだから，はっきりいってまだパソコンには荷が重い。HASHは思い切った簡略化で実験的に半影処理を達成した希有のシステムである。作成される映像は従来のレイトレーシングの弱点を見せつける。

が，HASHは使用するプリミティブを球に限定したことを抜きには成立しえないといっていい。まさに実験的であるにもかかわらず，完成されたバランスを持っているので次の一歩が踏み出しにくいのだ。

実用的なシステムを組むための方法としてはプリミティブを追加することではなく，影の計算と物体の計算を分離することが最善だろう。たとえば，プリミティブを使って物体を作成するときにその物体の内部を球で埋める処理を加える。レンダリング時の隠面処理はプリミティブを使い，影の計算部は球を使う……などが実現できれば実用に足るシステムを構築できると思われる。こうして次の実験が必要になってくる。

統合化へ

最近のグラフィックソフトの流行では2Dツールでも3D的な処理が行えるようになっ

Z'sTRIPHONY DIGITAL CRAFTでの作業例。マウスで部品を置いていくだけで手軽に3D CGが扱える。

2つの画像にそれぞれ処理を加え，さらにマスキングを施して合成したところ。Zs-EXの比較的単純な応用例だ。

一目瞭然の柔らかな影。右側が従来のレイトレーシングによるもので，左側がHASH.Xによる画像の出力例だ。

てきている。球や円錐などの物体を作成してその上にマッピングを施したり，できあがった画面を3D状に処理したりできる。Z's-EXで行っている3Dマッピングは一種のローカルレイトレーシングである。

また,3Dグラフィックがアカデミックだったころには御法度だった（?）3Dツールの出力をペイントソフトで修整するといったことも行われるようになっている。

もともと2Dも3Dも絵を描くための手法にすぎない。グラフィック環境は統合されるべきだろう。といっても全部まとめてしまえというのではない。必要もない機能までつけて重いシステムを作ってもしかたがない。統一する必要のある部分，それは第一にユーザーインタフェイス，次にデータ形式だ。データ形式は統一しなくても形式さえわかれば変換できる。しかしわからない（調べればすむという話もある)。たとえばZ'sTRIPHONYのデータ形式，Z'sSTAFFのアウトラインフォントフォーマット，サイクロンのデータフォーマット，CANVASのデータ構造……。

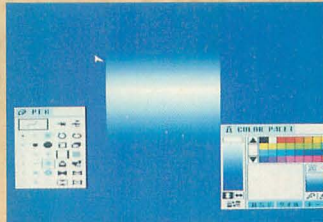
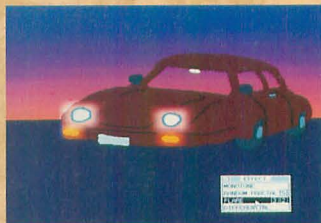
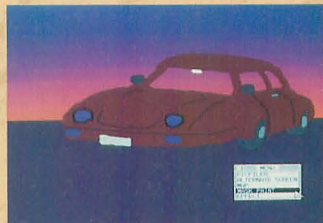
特にZ'sTRIPHONYのデータはサイクロンで扱え，なおかつZ'sTRIPHONY自体のエディタが非常に優れていることもあり非常に惜しい。本来ならポリゴンだけでなくレイトレ用のプリミティブも編集できるようにしてほしいところだ。いつだって強力なエディタはシステムの核となりうるのだ。

2Dに話を移そう。

X68000では画像データフォーマットとしてはPICが事実上の標準となった感がある。32768色のデータを実に見事に圧縮するPIC.Rそして拡張されたAPIC.Rはグラフィックを使う者にとって最高の福音だった。

今後はもっと嵩張るデータ，すなわちレイトレーシングのフルカラー出力（RGB各8ビット）に対しての拡張が行われるべきだろう。65536色の範囲で計算していてはマッハバンドは追放できない。1600万色で計算しておいて65536色に落とせばずっと綺麗な画像が得られる。ただ，こうしてできた画像は非常に圧縮効率が悪くなる。PICをもってしてもだ。24ビットデータ時点でPIC圧縮をかけ，展開時に65536色にするというわけだ。

すでに画面解像度の変換（拡大/縮小）や階調落としの技術は完成されているといっていい。展開に時間がかかることさえ厭わなければ非常に大きな意味での画像ファイルフォーマットの統一すら可能だろう。MacintoshIIやPC-9801のフレームバッファを使用し

Zs-EXで拡張されたさまざまな機能を使用したところ。マスクペイント，フレア処理，ランダムフラクタル，そして3Dマッピング……。

Z'sTRIPHONY用データ集「3D倶楽部」の使用例。ポリゴンでもこんなにリアルなCGが可能。気分はもうインテリアデザイナーだ。

ているユーザーは膨大なデータをもてあましているのだから。

SX-WINDOWのような環境であればもっとデータ運用を意識したツールも出てくるのだろうが，現状では閉じた世界が多すぎる。SX-WINDOW（正確にはSXシェル）のグラフィック能力がそれらを取り込めるほど（物理的に）高くないのが残念ではある。

DTPがますます流行し，今後は文書までも統合したソフトウェアが登場してくることだろう。そのときこれまでのグラフィックツールはどのように位置付けられるのだろうか。グラフィックを始め，さまざまなツールはどんどん広がりを見せる。そして，独自の世界を作ってしまう。

「拡張と統一の両立」，それがOh!Xの“実験”である。

製品一覧　X68000のグラフィック環境

●2Dペイント系ツール

数が多いので省略。詳細は1990年8月号のグラフィックツール紹介記事を参照のこと。

●2Dドロー系ツール

・CANVAS PRO-68K（シャープ）

カラー対応のポップアートツール。テキストセル，ドローセル，ペイントセルの3枚のセルを持ち，そのそれぞれにオブジェクトと呼ばれる部品を張り付けて絵を作り上げる。

ドローオブジェクトを制作するのには多少の技術と根気が必要なので，ドローグラフィックライブラリも用意されている。

●3Dポリゴン系ツール

・Z'sTRIPHONY DIGITALCRAFT（ツァイト）

ポリゴン（多角形）のモデリング/レンダリングシステム。三面図や透視図のさまざまな表示モードを使ってポリゴンを編集する。ポリゴンは単体だけでなく，いくつか集めたものをひとつのオブジェクトというように階層化して管理できる。回転体などの規則的な形状をしたオブジェクトは自動生成できる。

レンダリング時は，フラットシェイディングで通常のポリゴンとしての表現もできるが，スムーズシェイディングをかけることで，疑似的に曲面に見せることもできる。また，半透明なポリゴンの表現も可能である。

用途を限れば，データ集も用意されており，モデリングの手間を省くことができる。

●3Dレイトレーシング系ツール

市販されているものは2つ。両者に共通する部分を先に述べておくと，プリミティブ（基本立体）として球や円錐などの2次曲面や直方体などが使えること，プリミティブどうしの論理演算が可能なこと，反射・屈折および影の表現ができること，レンダリングの高速化を図るためにボクセル分割を採用していること，テクスチャ（カラー）・バンプ（凸凹）・アトリビュート（反射率など）のマッピングが使えること，物体をマクロ化して扱えることが挙げられる。

・C-TRACE68（キャスト）

ごくオーソドックスなレイトレーシングツール。

モデラーは付属していないので，形状ファイルはテキストエディタで作成する必要がある。

トランスピュータ対応版（C-TRACE TP）が用意されている。通常のものとデータの互換性がある。近くメタボールをサポートした新バージョン（C-TRACE+）がリリースされる予定である。

・サイクロンExpress α68（アンス・コンサルタンツ）

階層化モデラーというウリを持つレイトレーシングツール。モデラーでは，レンダリングに必要なすべてのパラメータを取り扱うことができる。マウスには対応していない。

ポリゴンを取り扱うことができる。これにより表現力が上がる。ただし，サイクロン単体ではだめで，ポリゴンのモデリングはZ'sTRIPHONY DIGITAL CRAFTで行い，それをサイクロンにデータコンバートして実現する。

入門者のためのグラフィックあれこれ

CGの基礎知識

Tan Akihiko 丹　明彦

CGにもいろいろありますが，いちばん問題になるのはどうやって絵を描くかということです。たくさんのアルゴリズムが生まれてきました。ここではそれらを理解するための基本的な考え方を紹介しましょう。

どうもグラフィックというと，ウケがよいわりに「数学が必要」で「難しい」というイメージができてしまった感がある。グラフィック特集のたびに違うことをやっているので新しい読者にはついてこられない面もあるだろう。

グラフィックを描くだけなら理屈はいらない。可能性を追求すれば必要になる。コンピュータグラフィックの歴史はまだ浅いが世界中の頭脳を結集したアルゴリズムが日進月歩で生み出されており，我々はその多くを適用できるハードウェアを目の前にしているわけだ。もうやるしかないではないか。

ここではこれまで扱ってきた，またはこれからも使われるであろう，さまざまなコンピュータグラフィックの用語や概念についてまとめを行っておく。まだ用語に慣れてない方やずいぶん昔のことなので忘れてしまっている方は特に注意して読んでほしい。ここでの解説では数式は一切出てこないので安心だ。

まずはCG(コンピュータグラフィック)の基礎の基礎から始めよう。

パソコンのグラフィックはディスプレイのドットの明るさを調整して図形を表現する。

色を表す場合には色を赤緑青の3原色(光の)に分解して表すことが多い。X68000の場合，それぞれ0～31の32段階が扱える(最大時)。これで表現できる色数は32×32×32＝32768色となる。

これを512×512＝262144個指定すればどんな絵でも描ける。しかし，これを手作業で指定することは512Kバイトのマシン語プログラムを作ること以上に困難なことだといえる。そこで効率のよいツールを使ったり，コンピュータに計算させたりするわけだ。それがCGである。

ひと口にCGといってもいろいろな形態が考えられる。ふつうに考えて，CGのもっとも大きな区分は平面(2D)か立体(3D)かということだろう(昔，桒野雅彦氏がやった4次元グラフィックとか祝一平氏がやった0次元グラフィックとかはふつうとはいえない)。もっとも，たいていは画面に出力されるわけだから，最終的に2Dには違いない。ここでの区分は内部処理やデータの構成によるものである。

2Dグラフィック編

なにが2次元かというと，画面が2次元だからそれにあわせて絵を描くと2次元になるというわけだ。もともと絵というのは平面的なもの。感覚的にもいちばんわかりやすいのではないかと思う。

ここで出てくる用語として「ビットマップ」というものがある。ほとんどのグラフィック画面はビットマップだから気にする必要のない単語ではある。画面上のドットに任意に色指定できること，といった意味で捉えればいい。要はふつうのグラフィック画面のこと。X68000の画面などはワードマップといったほうがいいハード構成なのだが。

ビットマップでないものの代表例がX68000のスプライトBGやX1のPCGだ。スクロールゲームの背景などの多くは四角形のチップで構成されている。限られた色数と限られたチップ数で絵を描く技術，ソフトハウスのチップワークの優劣はゲーム画面を見ていると明らかだ。うまく運用するには独特なテクニックがあるらしいが，詳しいことは割愛する。

ペイントとドロー

2Dグラフィックツールを見てみよう。Z'sSTAFFなどの一般的なグラフィックツールはペイント系のツールに分類される。CANVASなどはドロー系ツールだ。こういうと難しそうだがペイント系はごくふつうのツール，問題はドロー系だろう。ドローは絵を構成する要素を手順で示し加工することで絵を描いていく。

極端にいってしまえば，ペイント系は画面上のピクセルを直接いじるツールである。ピクセルは，日本語では画素という。画面上のドットのことである。X68000の65536色モードでいえば，512×512個のドットがあり，そのひとつひとつをピクセルと呼ぶ。で，そのピクセルの色をひとつひとつ決めるのがペイント系のグラフィックツールなのである。ただ点を打っていくのではあまりにも面倒だし，コンピュータの上に載せる意味がないので，さまざまなペンや便利な編集機能を載せているのである。

対してドロー系の立場はかなり異なる。直線や曲線や多角形といった図形を部品として捉えている。文字も部品のひとつである。その部品を1つひとつ紙の上に乗せていく，というのがドロー系のグラフィックツールのイメージといえる。紙の上に乗せるだけなので，あとから修正もできるし，取り除くこともできる。拡大・縮小・変形も，部品の座標データをいじるだけなので容易に行うことができる。

半面，部品自体が単純なものになりがちで，絵画的なものを描くのは不向きといえる。図版などを描くには非常に強力。ちなみに今回の図版にはX68000用のCANVAS PRO-68Kを使用している。

3Dグラフィック編

3次元グラフィックはパソコンの醍醐味を教えてくれる。2Dならばコンピュータでなくても可能だが，3Dとなるとコンピュータを使わなければちょっとできないことだ。

ここでいう3次元とは内部処理のことを指す。立体視とか赤青メガネとか，本当に

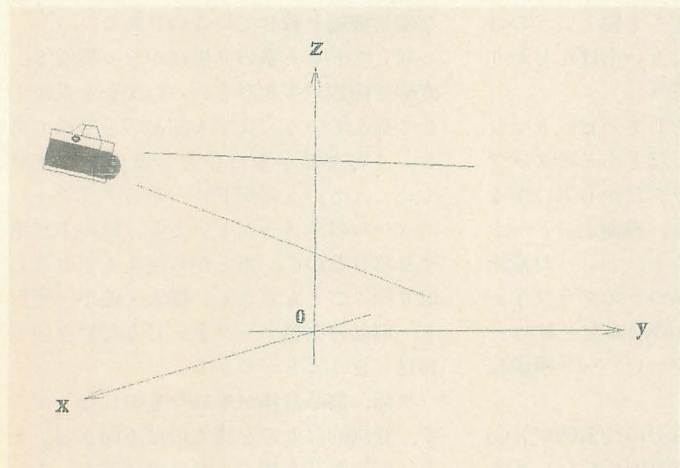
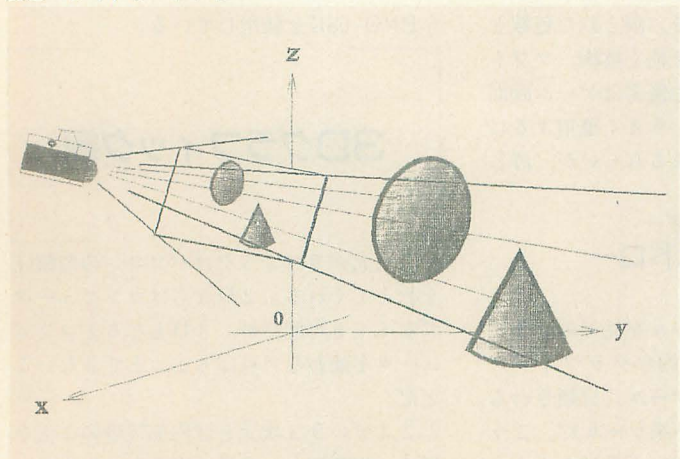
立体に対応するアプローチもあるが，ここでは単に３次元座標でデータを扱うもの，要するに遠近法などを自動で処理してくれるものと見てほしい。

隠面処理

3Dコンピュータグラフィックとなると，問題になるのは，物体の前後関係である。２つの物体がある。２つは重なって見える。ところが，遠くにあるはずのものがもうひとつのものを隠しているように見えたら，たちまちにして両者の前後関係を認識することは不可能になってしまう。遠くにあるものは近くにあるものに隠される。これがあるべき姿である。

この隠面消去と呼ばれる問題には解決法がいくつかある。そのどれにも共通しているのが，結局近くにあるものを描く，というごく当たり前のことである。

まずスキャンライン法とｚバッファ法から。両者は違う方法ではあるが，似た部分はある。

構図はユーザーが決めている。ひとつのシーン内には物体がいくつもあり，その中にある位置と角度でカメラを置く，というイメージである。

構図が決まったら，ここからは計算機の仕事である。処理をしやすくするために，カメラの向きがまっすぐになるようにシーン内の物体を移動してやる（少々不正確ないい方だがお許しいただきたい）。まっすぐ，とは，ｚ軸に平行，という意味である。こうしたおかげで，物体をｚ座標の順に並べるだけで，物体を視点に近い順に並べたことになる。そして，視点から遠い物体から順に画面に置いていけば，結果として，重なった部分では近いものが見えるというわけ。

これをもう少し高度にしたものがスキャンライン法やｚバッファ法と呼ばれる描画アルゴリズムである。

そしてレイトレーシング。レイトレーシングといえば派手な反射や屈折が表に出てきがちだが，もともとは隠面消去の一手法である。

レイトレーシングでは，シーン内の物体をカメラの正面に持ってくるということはしない。カメラから視線を無数に発生させてシーン内に飛ばす。そして，最初にぶつかった物体がカメラから見えているということにするのである。カメラのフィルムを画面になぞらえ，画面上の１点１点に対応する視線をシーン内に飛ばし，最初にぶつかった物体の色を画面に書き込めば，それで隠面消去ができることになる。

レイトレーシングの考え方は，そのまま反射・屈折，それに影の表現に応用できる。ぶつかった物体がもし鏡のようなものであった場合，その視線が鏡に反射してどの方向に行くかは簡単に計算できる。したがって，このぶつかった点を新たな視点として新しく視線を飛ばし，どの物体にぶつかるかを調べる。これが反射の原理である。

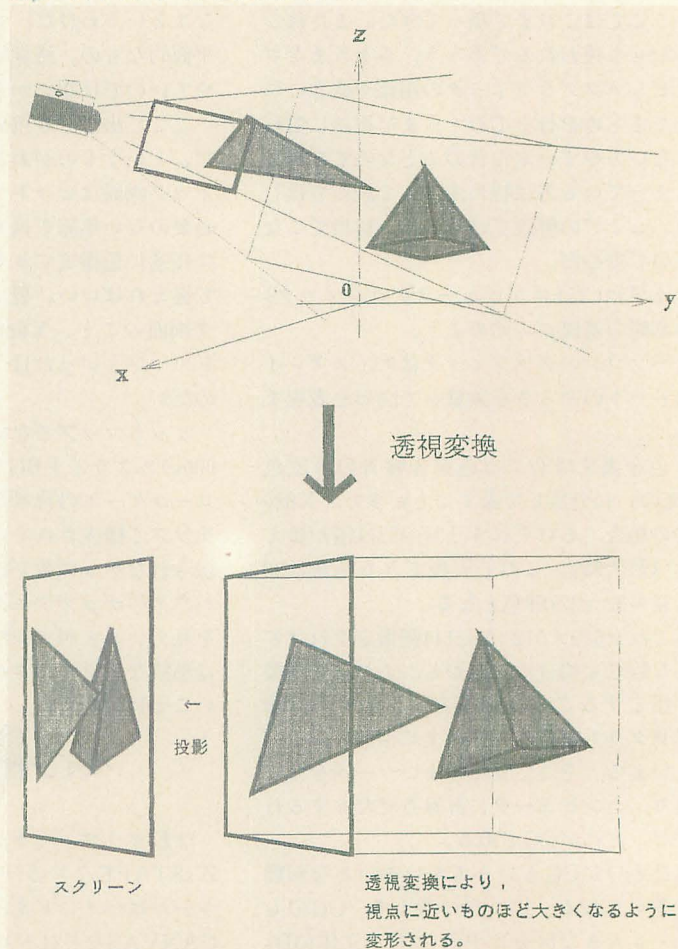
屈折も同様である。

影であるが，視線がぶつかった点を新たに視点とし，今度は光源に向かって視線を飛ばす。もし，光源にぶつかる前に他の物体にぶつかったら，最初にぶつかった点は実はその物体の影に隠れているということがわかるのである。したがって，そこは影になる。

図1　シーン

図2　レイトレーシング

図3　スキャンライン法またはｚバッファ法

図版はCANVAS PRO-68Kで作成

3Dグラフィックの2大潮流

ここまでの話からもわかるように，3次元コンピュータグラフィックは大きく2つの流れがある。その2つの流れとは，レイトレーシング法を用いたものと，スキャンライン法やzバッファ法を用いたものである。それぞれ，ここでは「レイトレーシング系」と「ポリゴン系」と呼ぶことにする。実はこの呼び方は大間違いなのだ。なぜ間違いかというと，レイトレーシングでもポリゴンは扱えるから。しかしわかりやすさを優先することにしよう。

念のためにいっておくと，ポリゴンとは「多角形」のことである。多角形といっても実際に使われるのは三角形，せいぜい四角形がほとんどで，ポリゴンの3Dグラフィックというのは三角や四角をツギハギして立体を作ることだと思っていい。

速度について

レイトレーシングは，いまのところパーソナルコンピュータにおいてはかなり高い地位を占めている。しかし批判も多い。曰く「遅い」「リアリティがあるというがそれには制限がある」「形状表現の自由度が低い」などなど。

これらはいずれも，レイトレーシングの持つ弱点を突く，痛い批判である。逆にポリゴン系はこれらの弱点は克服している。しかし，それでレイトレーシングの存在価値が失われるわけではない。ポリゴン系にもポリゴン系なりの弱点がある。両者は共存可能なものなのだ。

描画速度について考えているうちに，レイトレーシング系とポリゴン系の違いがひとつ浮かび上がる。

レイトレーシングの場合，計算機は「いま，なにを描画しているのか」を知らない。視線を発生させて空間に飛ばし，シーン内のすべての物体と交差判定を行い，その中でもっとも近い物体を決定し，そこで初めて描くべき物体を知るのである。これをピクセルごとに繰り返す。

ポリゴン系の場合は事情は逆で，登録されている物体をひとつずつ描画していく。いい換えれば「いま，なにを描画しているのか」を初めから知っているのだ。

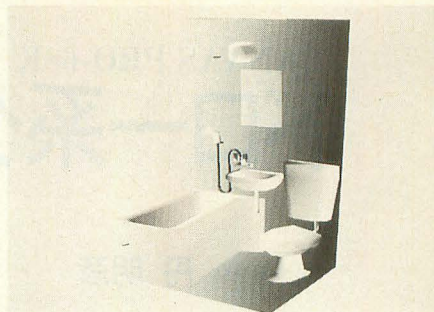
この差は大きい。レイトレーシングが遅い遅いと悪態をつかれているもっとも大きな原因はここにあるのだ。なにを描くべきか知らないのだから，ピクセルの個数だけいちいち探し回ることになる。とんでもないところにある物体でも，一応は視線と交差しているかどうか検査しなくてはならない。事実，レイトレーシングの処理時間でもっとも大きなウエイトを占めているのが交差判定で，全体の数十%，ときには90%以上にも及ぶ。

これを軽減するために，ボクセル分割など数々の工夫がある。なにがどこにあるかを前もって調べておけば，探し回る手間を大幅に省略できる。

CGのリアリティとは

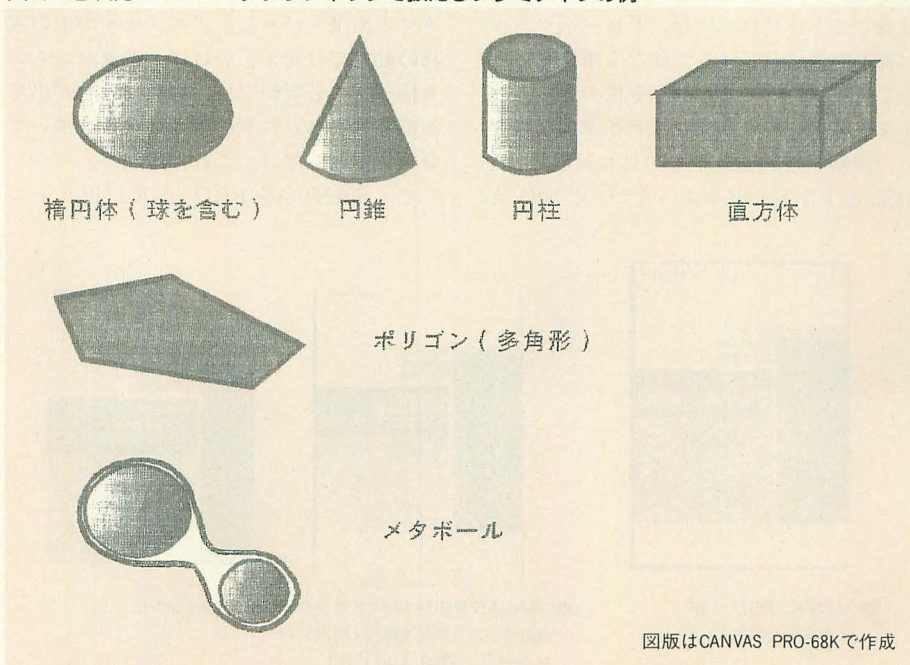
半分嘘混じりなのを承知でいうが，ポリゴン系は形状の表現に関して自由度が高く，レイトレーシング系は質感の表現に対して自由度が高い。

レイトレーシングは確かに写真のようにリアルな画像を出力することができる。しかし，現実とまったく同じリアリティを得ることができるかというと，決してそんなことはないのである。レイトレーシングの場合は，使えるプリミティブが限られており，もしそれをちょっとでも外れるようなものを作ろうとすると，たちまちにしてリアリティは崩壊してしまうのだ。どこのメーカーのクルマでもいいが，レイトレーシングで作ってみるといい。写真のような作品ができるだろうか。たぶん，とてもみっともないものになるであろう。

レイトレーシングで扱えるプリミティブは，実現が簡単ということもあり，多くは球などの2次曲面にとどまっていた。そして，それらを論理演算で組み合わせたり切り取ったりして目的の形状に近づけていた。したがって，「チェック模様の台に乗った金属球」とか，「ガラスのテーブルに乗ったワイングラス」ならば，写真と見まごうばかりのリアルな画像を出すこともできるが，自動車のような複雑で微妙な曲面を組み合わせた形状は苦手なのである。ここに，レイトレーシングの表現力のひとつの限界があった。

ポリゴンによる立体表現(Z'sTRIPHONY)

ただし最近になって，パーソナルコンピュータ上でも自由形状を扱えるレイトレーシングソフトが現れつつある（プロの世界ではずっと前からあった）。ポリゴンやメタボールである。いずれも2次曲面に比べて重い処理であるが，とりあえずスキャンライン法などを用いたポリゴンレンダラと肩を並べるだけの表現力は身につけた。反射や屈折や影を簡単に表現できる分，ポイントは高い。

いつも問題は速度だけだ。

図4 3次元コンピュータグラフィックで扱えるプリミティブの例

図版はCANVAS PRO-68Kで作成

CANVAS PRO-68K
ドロー系グラフィックツールの魅力

Tan Akihiko 丹 明彦

Macintoshの世界では多くのドロー系グラフィックツールがありますが，国内にはそういったツールはほとんどありません。そこに登場したのがX68000用のCANVAS PRO-68Kです。データ集も出たところで改めてその魅力を探ってみましょう。

前回の紹介からしばらく時間がたっている。その間に，CANVAS PRO-68Kは正式に発売され，はたまたデータ集も発売され，もうお使いの方も多いことと思う。そこで第2回のレポートをお届けする。

前回のおさらいから

CANVAS PRO-68Kは，いわゆるドロー系のグラフィックツールである。これまでX68000のグラフィックツールといえば，Z'sSTAFFを始めとするペイント系のツールであった。ペイント系のツールは，画面の上に色を置いていく。対してドロー系のツールは，部品を作って，画面の上に部品を置いていく。その部品はいつでも修正できる。

CANVAS PRO-68Kは図形の印刷を主眼としたツールである。プリンタに出てくるイメージそのままの画面で編集することができる，いわゆるWYSIWYG方式のポップアートツールなのだ。

美しい印刷ができるグラフィックツールは長いこと待たれていた。ドロー系ツールは部品の情報をドットではなく座標という形で持っているので，部品を拡大してもペイント系ツールのようなモザイクにならない。印刷を目的としたものにはNEW PrintShop PRO-68Kがあったが，CANVAS PRO-68Kはドロー系のツールであるということをセールスポイントにしている。

基礎知識編

前回は技術的な部分にまったく触れなかった。その中で基礎知識として説明しておきたいことがいくつかある。ペイント系のツールに慣れた人には馴染みのない概念も多いことだろう。

CANVAS PRO-68Kはセルを編集する

CANVAS PRO-68Kにはたくさんの動作モードがある。それを押さえておこう。おいしい部分を味わうのはそれからだ。

アニメセルというのをご存じか（なんて失礼な質問)。アニメセルは透明なシートで，それに人物の絵を描く。あらかじめ描いておいた背景の上にアニメセルを重ねれば1コマの出来上がりだ。アニメセルの上の人物の動きだけ変えていけば，背景をいちいち描き直す必要がない。複数の人物が出てきても，それぞれ別のアニメセルに描いておいて重ねるだけ。これは便利だ。

で，CANVAS PRO-68Kが扱うセルはこのアニメセルに性格が似ている。セルは3種類で，それぞれ，

- ・ドローセル
- ・ペイントセル
- ・テキストセル

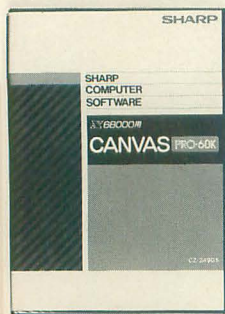
と名前がついている。この3枚のセルを重ねて1枚の作品を作る。3枚の間の優先順位は自由に変えられる。

さて，この3種類のセルの上で編集する部品はそれぞれ違ったものである。

ドローセルはCANVAS PRO-68Kの目玉である。直線はもちろん，自由曲線，円，多角形，その他さまざまな図形を扱える。何度もいうように，これらは印刷する瞬間まで直線や曲線といった部品としての属性を失わないので，プリンタに出力したときの図形は，あくまでプリンタの解像度をめいっぱい生かしたものになる。ハードコピーとはこの点で違う。一般にプリンタのほうが画面よりはるかに解像度が高いので，ハードコピーした画面では粗さが拭えない。その点，ドローセルは座標データから印刷パターンを起こすので，いくら拡大しても大丈夫。

一方のペイントセルは，通常のペイント系ツールに似ている。グラフィック画面に図形を塗り付けてセルを作り上げる。ただこのセルだけが唯一画面のイメージそのものをデータとして扱うので，解像度の点では少し弱い。

そしてテキストセルは，文字を扱う。文字はいろいろな大きさで描けるし，自由な配置も可能である。ベースラインを曲線で指定すれば曲がった文章も描ける。小さな文字をびっしり並べてスクリーントーンのようにすることもできる（かなり重くなるが）し，大きな文字を出すこともできる。

そしてなによりも違うのは，こうして作った文字の1つひとつはセルの上に浮いている部品なのだ。まだセルの上に張りついているわけではないので，マウスでつかんでずりずりと動かすこともできる。指定した曲線の上にくねくねと並べることもでき

●CANVAS PRO-68K
29,800円（税別）

●CANVAS PRO-68K ドローグラフィックライブラリ
Vol. 1，2　各8,800円（税別）
シャープ　☎03(3260)1161

る。さらに，変形することも簡単にできる。文字の装飾もいろいろあって，白抜き文字や太文字，影つき，斜体，などが使える。

半角フォントに関しては，CANVAS PRO-68Kのシステムディスクにいろいろなものが用意されている。これはアウトラインフォントなので，拡大しても大丈夫。印刷しても文字の品質は落ちない。

全角フォントは，デフォルトではROMフォント（にスムージングをかけたもの）だけだが，Z'sSTAFFのバージョン2などについてくるアウトラインフォントなど，品質の高いものを利用することができ，巨大な明朝体の文字も，楽に美しく出せる。

スケルトンモードとリアルモード

ドローセルとテキストセルには，それぞれスケルトンモードとリアルモードというモードが用意されている。

スケルトンとは骨格のこと。スケルトンモードでは，編集する図形の骨組みを，細線で表示している。出来上がりのイメージとは多少ずれるのだが，表示の速度は速い。通常，編集作業はこのスケルトンモードで行う。ドローセルでは，ベジェ曲線の制御点はこのモードでのみ表示され，移動や削除もこのモードで行う。

リアルモードは，指定した色や装飾などをすべてつけて画面に表示するモード。プリンタに印刷するときのイメージそのままである。印刷前に最後のチェックをするためのモードと考えていいだろう。

要するに，スケルトンモードは「作る」ためのモード，リアルモードは「見る」ためのモードである。

編集モードと浮遊モード

テキストセルにはスケルトンモード/リアルモードのほかに，編集モード/浮遊モードもある。

テキストセルの編集モードは，いってみればテキストエディタ。ここでどういう文章を入れるかをキーボードから直接打ち込む。文字の色や装飾，フォントの種類や大きさの指定はこの編集モードで行う。これらの指定は文字単位で行えるので，いろいろと凝ったこともできる。1文字ごとにフォントを変えたら，脅迫状だって作れるぞ(冗談)。

で，文章を作り終わったらおもむろに浮遊モードに移り，思いどおりのレイアウトになっているかどうかを見る。編集モードはあくまでテキストエディタなので，実際の印刷イメージはこの浮遊モードに来てみないとわからない（さらに色や装飾の様子を確かめたいときは，スケルトンモードからリアルモードに移行する)。うまくいっていないときは，また編集モードに戻って作り直すこともできる。ここが，文字をも部品として扱うことのできるドローツールの強みである。

浮遊モードでは，文字をあれこれと動かすことができる。文章の下にベースラインと呼ぶ線が現れる。それをつかんで動かせば，その文章を画面の好きな位置に移動することができる。ベースラインを変形する（自由曲線になっている）ことで，曲がりくねった文章を作ることもできる。ただし，文字そのものを変形することはできない。変形したいときは，前回のレビューでやってみせたように，ドローセルにコピーして，ドローオブジェクトとして変形する（このとき，文字としての属性は失われているので，たとえばフォントを変更したいと思っても手遅れである)。

ペイントセルにはスケルトンモード/リアルモードはないが，編集モード/浮遊モードはある。編集モードでは円や長方形などのペイントオブジェクトを作る。ペンを使って細かい修正をすることもできる。

そのあと浮遊モードに移って印刷イメージを見たり，ペイントオブジェクトを動かしたりできる。むろんペイントオブジェクトも部品であるから，編集モードに戻って自由に作り直すこともできる。

具体的にはなにができるのか

さて，CANVAS PRO-68Kにはいろいろなセルと動作モードがある，といってみたところで，具体的にはなんのことかわからないだろう。

CANVAS PRO-68Kの作業は，

・部品を選択する。
・道具（編集項目）を選択する。
・部品を加工する。

といったことを繰り返しながら進む（ドロー系ツールだからこそ，わざわざこんな言い方をして強調するのだ)。この部品の種類も多いなら，道具の種類も多い。当然，道具はセルごとに違う。ちょっといじったくらいで全貌がつかみきれるものではない。

ドローセルで使える道具のうち，いくつかについて解説しよう。

●ペン

折れ線または曲線を描く。マウスをクリックすれば折れ線の頂点に，ドラッグすればベジェ曲線の制御点になる。ドラッグしている間，ベジェ曲線はマウスの動きにつれてくねくねと動く。これがけっこう楽し

ベジェ曲線

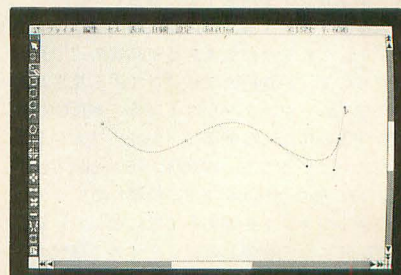
CANVAS PRO-68Kで自由曲線の登場する場面はいくつかある。ドローセルでは，たとえば自由曲線そのものを描くとき。またドローオブジェクトをエンベロープ変形するとき。テキストセルでは，ベースラインを変形して，文字の曲がりくねった置き方をするのに使っている。CANVAS PRO-68Kでは自由曲線にベジェ曲線を用いている。

ひとつのベジェ曲線を発生するためには，制御点が4つ必要である。ここからがちょっとわかりにくいのだが，曲線が通るのは1番目と4番目の点だけで，2番目と3番目の点は通らない。この2つの制御点は，「曲線を自分のほうに引き寄せる」働きをするものと考えていいだろう。

どうしてこんな難しそうな話をするのかというと，CANVAS PRO-68Kで自由曲線を操作する際には，この4つの制御点を直接操作する必要があるからである。別に理論など知らなくても，しばらく使っているうちに操作感覚（結構楽しい）はつかめるのだが，知っておくと操作法の理解も早いことだろう，というわけでおせっかい。

本文でも書いているが，操作法は極めて簡単。適当な場所にマウスを持っていき，そこでマウスのボタンを押してドラッグすればベジェ曲線が出現する。何度も繰り返せば，次々とベジェ曲線が接続していく。出来上がりはマウスのボタンを押した点を通る自由曲線になる。制御点の数はできるだけ少なくしたほうが美しい曲線になる。コツはすぐにつかめることだろう。

一度描いたベジェ曲線を修正するときも，曲線をつつけば制御点が出現するので，それをマウスでつかんでドラッグすれば曲線が面白いように動く。

百聞は一見にしかず，ぜひ一度触ってみることをお勧めする。思いどおりの曲線を描けるようになるには少々時間が必要かもしれないが，慣れると快感である。少ない手間で美しい曲線を作れるベジェ曲線は，ドロー系ツールの必携アイテムといえるだろう。

く，ベジェ曲線を描いているだけでも飽きない。折れ線と曲線を，モード切り替えなどしないでごく自然につなぐことができるというのが賢い。

●ポインタ

ドローオブジェクトを指定したり移動したりする。基本操作は編集したい点をクリックすること。矩形領域をドラッグして囲めば，領域内の点をすべて選択する。1点だけ指定すれば，その点に付属するベジェ曲線の制御点が現れる。制御点をマウスでつかんでドラッグすれば，ベジェ曲線もそれにつれて動く。気に入った形になるまであれこれとドラッグしていればいい。くどいようだが，いったん画面上に置いたあとでも，修正を加えることができるのがドロー系ツールの強みなのである。

ほかにも，OPT.2キーを押しながら制御点をクリックすると，その制御点を含むオブジェクト全体が選択される。そこでおもむろに制御点をマウスでつかんでドラッグすれば，オブジェクト全体を動かすことができる。

●長方形

長方形オブジェクトを作る。左上の座標と右上の座標を指定する。長方形といっても，プログラム上の扱いは4本の線分にすぎない。つまり，頂点をマウスでつかんでドラッグすれば，好きな形の四角形が作れる。

●円

円オブジェクトを作る。中心と半径を指定する。やはりこれも4本のベジェ曲線がつながっているものなので，あとから変形することも可能。正多角形もある。

●回転

指定したオブジェクトを回転させる。回転の中心と回転始角と回転終角を指定する。上下左右反転もある。

●エンベロープ変形

もっとも遊べてしまう道具がこれ。指定したオブジェクト全体を包む大きさの長方形をエンベロープとし，これに拡大縮小，変形，ベジェ変形（縦変形と横変形とがある）といった変形を加えることができる。

●パスクローズ

耳慣れない名前だが，頻繁に使う。ペンで描いた曲線（または折れ線）は1本の曲線で，どう制御点を動かしても円や長方形のようにループを作ってはくれない。この閉じていない曲線の端点をくっつけて閉曲線にしてくれるのがこの道具である。この処理をしておかないと，オブジェクトは面として働いてくれない（色をつけることができない）。

●オブジェクト属性設定

オブジェクトの輪郭線の色や太さ，それにオブジェクトが閉領域だった場合（そうでない場合は上のパスクローズを使って閉曲線にしておかないと色がつかない）は面の色を決めることができる。色は，8色だけでなく，タイルパターンから選ぶことができる。さらに面に色をつける場合は，グラデーションまでかけることができる。

ちなみに色であるが，8色と書いたのは実は不正確である。CANVAS PRO-68Kは基本的には8色だが，グラフィック16色モードで動作している。しかし残りの8色も決して無駄にはなっていない。残りは透明色に当てられている。つまりCANVAS PRO-68Kの扱える色は不透明色8色，透明色8色，の計16色なのだ。これは重ね合わせのときに威力を発揮する。CANVAS PRO-68Kでは，セルどうしまたはオブジェクトどうしを，優先順位をつけて重ね合わせることができる。このとき，上のセルまたはオブジェクトに透明色が使ってあると，下に隠れたものが透けて見えるのである。つまり，透明色と不透明色を使い分ければ，マスキングの効果が出せるのだ。

＊

道具についてはこれくらいでいいとして，編集コマンドについても触れておかなくてはなるまい。編集メニューからマウスをクリックして（またはOPT.1＋しかるべきキーを押して），編集コマンドを呼ぶ。ドローセルだけでなく，ほかのセルでもこの編集コマンドは使えるのだが，とりあえずドローセルの話と思うことにする。

●カット/コピー/ペースト

説明するまでもないだろう。これが簡単にできるのもドロー系ツールの強みのひとつ。

●グループ化

ドローオブジェクトをいくつか作ったとしよう。それらをひとつのオブジェクトとしてまとめて扱いたくなる場面は必ず生じる。たとえば移動するとき。全部いっしょに動いてほしいものだ。しかし別々のオブジェクトのままでは，なにかと厄介。そんなときは，あらかじめグループ化しておくのだ。元の別々のオブジェクトに戻したいときには，逆の動作をする「アングループ化」も用意されている。

●オブジェクトの上下関係を変える

先ほども述べたように，CANVAS PRO-68Kは不透明色と透明色を使い分けることができる。不透明なオブジェクトをかぶせて下のオブジェクトを隠すこともできる。それには，オブジェクトの優先順位をコントロールすることがどうしても必要になるわけだ。これはそのための編集コマン

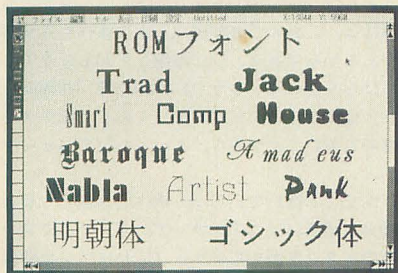
さまざまなフォント

CANVAS PRO-68Kには，いくつかの半角文字のアウトラインフォントがついてくる。どれもが「ポップな」フォントなので，英数字でロゴタイプを作るときにそれほど困ることはないであろう。付け加えておくと，このベクトルフォントは，NEW PrintShop PRO-68Kに付属しているものと同じものである。

半角は文字の絶対数が少ない（アルファベットの数なんて漢字に比べたらものの数ではない）ので，いろいろなデザインのフォントが比較的簡単に供給できる。それに対して全角フォントを取り巻く状況は厳しい。漢字というのは計算機にとって厄介なものなのである。数千字にもおよび，しかもアルファベットよりはるかに複雑な漢字のフォントに対して，種々のデザインのフォントを用意するのは苦しい。

とりあえず使えるのは内蔵ROMの24ドットフォント。これをただ拡大すると品質の点で見劣りするので，一昔前のワープロでよく見られたようなスムージングをかけている。本格的なロゴを作るのにこんな貧弱な（失礼）フォントではだめだというので，CANVAS PRO-68Kでは，Z'sSTAFF PRO-68K ver.2.0に付属しているアウトラインフォントを使えるようになっている。Z'sSTAFFには明朝体とゴシック体の2種類のアウトラインフォントが付属している。これで当面は用が足りる。ただ，フォントのファイルは巨大で，フロッピーでの使用はまず無理。ハードディスクがないとつらいところ。

また，Z'sSTAFFに付属しているのは第一水準の漢字だけなので，第二水準の漢字を使いたい場合は，Z'sWORD JGに付属のフォントを持ってくる必要がある。

ほかにもCANVAS PRO-68Kで使えるフォントには，「書体倶楽部」というのがあって，ゴシック体や明朝体だけでなく，各種の書体が用意されている。この両者ともPC-9801シリーズ用ではあるが，フォントのデータは利用できる。ちなみにこれらのフォントは当然ながらZ'sSTAFFでも利用できる（そりゃそうだ，どれもツァイトの製品だもの）。

ド。指定したオブジェクトを一番上に置いたり，一番下に置いたり，ひとつ上に置いたり，ひとつ下に置いたりできる。

●複製

これまたドロー系の強み。指定したオブジェクトを複製する。特においしいのが「多重複製」というやつで，オブジェクトを一瞬にして何枚も複製し，しかも等間隔で並べてくれる。

*

とまあ，こんなところだろう。CANVAS PRO-68Kだとわかっていなかったら，とてもグラフィックツールの解説とは思えなかったことであろう。特にドロー系のツールらしい部分をピックアップして紹介したつもりだ。

使用レポート

それでは，実際にある程度使用したうえでのレポートに移ろう。もちろん，国内には比較対象になるようなソフトがほとんどない。こういったソフトが登場したこと自体は十分に評価に価すると思うが，それでもあえて，現時点で思いつく要望は残らず書くことにしよう。

CANVAS PRO-68Kの道具選択

CANVAS PRO-68Kの編集はマウスとキーボードを併用しながら進める。キーボードといっても，テキストセルの文字入力に使うだけではない。よく使う編集コマンド，たとえば「カット」や「ペースト」などは，キーボードにも割り当てられている。もちろんマウスだけでも編集コマンドは使えるが，キーボードが併用できると，いちいちマウスを画面上方に持っていく必要がないので，素早い操作ができ，けっこう便利。はっきりいって，付属ワープロにも欲しいくらいである（ちなみにHyperwordはこの作法を採用している）。

Macintoshを使ったことのある方なら，プルダウンメニューの中になにやらアルファベットが書いてあって，そのキーを押すとマウスでそこをクリックしたのと同じ機能が使用できるのをご存じだろう。あれと似ている。

メニューはプルダウンにちょっと似ているが，ちょっと違う。画面上端のメニューバーにマウスカーソルを運んで目的の項目でクリックすると，サブメニューがぽんと開く。プルダウンメニューはここからドラッグしてアイテムを選んでいたが，CANVAS PRO-68Kのメニューではドラッグしない。目的のアイテムのところでクリックするのだ。するとその項目が選択され，サブメニューもぱたんと閉じる。間違えたときは右クリックするか，またはサブメニューの外側でクリックするかすればサブメニューは閉じる。

このメニュー形式の操作性は上々である。プルダウンメニューに勝るとも劣らないものがある。プルダウンメニューでは，ドラッグの途中でマウスのボタンから指が離れてしまって妙な動作をするという事故がありがちだが，その心配はない。反面，メニューを閉じる動作が必要になるので，もしかするとうっとうしいかもしれない（うっとうしいと思ったことはないが）。さてあなたならどちらを選ぶか。

ドロー系ソフトの使い勝手

CANVAS PRO-68Kのポイントは1にも2にもドロー系グラフィックツールであるということだ。が，ペイント系ツールを使ってきた人にとっては，かったるいというのが正直な感想になると思う。確かに使っていてある種のまだるっこしさはつきまとう。

ここで誤解があっては困るのだが，CANVAS PRO-68Kそれ自身の操作性は悪くない。変に数字を意識しなくても使えるし，マニュアルなしでも大方の操作はできる。

しかし，ドロー系ツールは必ず「手順」というのを持ち，そのためにペイント系ツールに対して少なからぬハンデを背負っていることは否定できない。

ドロー系ツールで作った部品の持つメリットと，部品を作るための手間とを天秤にかけると，ドロー系にすべきかペイント系にすべきかを選択できると思う。おそらく，描く対象によって両者を使い分けることになるだろう。

簡単な図の類であれば，ドロー系を使うのが断然有利である。特にCANVAS PRO-68Kは文字も使える。線や文字の位置を微調整するのもドロー系なら楽である。

速度について

操作体系とは別の意味でもCANVAS PRO-68K はかったるい。処理速度が十分だとはいいがたいのである。リアルモードの描画が遅いのはある程度しかたないとしても，スケルトンモードの遅さはもう少しなんとかすべきではなかったか。

むろん，簡単な図形のうちはそれほど気にならない。が，ちょっと凝ったものを作ろうとすると制御点がどんどん増え，編集も描画も遅くなっていく。データディスクについてくる干支や恐竜やトランプなどのドローオブジェクトを読み込んで編集してみると，そのことを実感できる。アウトラインフォントの処理もけっこう重くなる。

複雑なものほど遅くなるというのは，グラフィックツールにとっては困りものである。ちゃんとした絵というものは，往々にして複雑な絵柄になるものだから，作業が佳境に入ってからの速度にはシビアなものが要求される。

ペイント系のツールでは，絵柄が複雑だろうと簡単だろうと，描画にかかる時間は変わらない。ドロー系のツールでは，ある程度遅くなるのは避けられないし，しかたないのだが，ユーザーにストレスを与えてしまうほど遅いものであっては困るのだ。

通常の絵を描くには，まだまだペイント系のツールのほうが使いやすいといえる。CANVAS PRO-68Kは，製図道具で描いたような正確な直線や曲線にむしろ向いているといえよう。

CANVAS PRO-68Kが遅いのは，CPUが10MHzの68000だからだとは思いたくない。確かに，X68000の処理スピードというのは今では決して速い部類ではない。特にグラフィックを扱うときに弱さを露呈するというのは否定できない。だが，X68000のグラフィックアプリケーションが遅いのにはそれだけでない，別の理由があると思う。なにかにつけてヘビーデューティなのだ。考えられるすべての機能を盛り込んだのではないかと思わせるものがある。しかも処理は生真面目にやっている。いつかも書いたが，場面によっては，「真面目」というのはほめ言葉でもなんでもない。馬鹿正直な処理は遅いだけであり，誰も認めてくれはしない。ユーザーの知らないところでどんなずるをしてもいいから，とにかく速度優先で作ってほしいものだ。どうせ相手はプリンタなのだから，いっそのこと白黒に限定するというのも手ではある（MacDrawと差別化できなくなるかもしれないが）。

外部とのインタフェイスが弱いぞ！

CANVAS PRO-68Kにはビットイメージそのものを編集するペイントセルがあ

るが，これの操作性はそれほどほめられたものではない。そこで，この方面ならZ'sSTAFFなどのペイント系ツールにお任せとなる。CANVAS PRO-68Kは，Z'sSTAFFの画像ファイル（ZIMファイル）を8色に変換してペイントセルに取り込むことができる（ZIMファイルのほかに，GL3ファイルやNEW PrintShop PRO-68Kのファイルを取り込むことも可能である）。

CANVAS PRO-68KとZ'sSTAFFは，正反対の性格を持っているだけに，組み合わせて使うとお互いの不足したところを補い合う名コンビになる。……はずだったのだが，これにはひとことといっておかなくては収まらないものがある。

なんなんだあの8色変換は。

いくらなんでもひどすぎないか。65536色が泣く。要するに，RGBの最上位ビットだけを残し，あとはバッサリ，なのである。取り込んだ絵ははっきりいって悲惨である。

ペイントセルが，印刷したときの解像度も粗く，ドロー系ツールの精神にマッチしていないことも考えると，あまり真剣にサポートしたくないというのもわかるのだが，もっと自然に変換しようと思っても，ほんのちょっとの手間ですむ。65536色モードの画像を読み込むことはわかっているのだから，ディザ（8色のドットをうまく混ぜ合わせて疑似的に中間色を出す仕掛け）をかけて取り込むモードをつけてもバチは当たるまい。

スキャナもサポートしているが，これも同様。やはり最上位ビットのみしか見ていない。ただしスキャナの場合には抜け道がある。たとえばシャープのスキャナであるCZ-8NS1の場合，スキャナのほうでディザをかけてくれる。スキャナがディザをかけるモードになっていれば，CANVAS PRO-68Kに取り込んだあともそれほどひどい画面にならずにすむ。しかしこれはあくまで対症療法，賢い生活の知恵にすぎない。だいたい，こんなことはソフトのほうで対処すべきことだと思う。

また，スキャナ機器との接続はRS-232Cのみで，パラレルボードは無視されているようだ。高価なパラレルボードを買った人は泣くに泣けない。いまどき，98ユーザーでもRS-232Cの転送速度で我慢できる人は少ないというのに……。

＊

それから出力である。めぼしいプリンタには出力できるようになっている。しかし，とりあえずはいっておこう。

レーザープリンタに対応しないのか？

ドットプリンタや熱転写プリンタの印刷はだいぶ綺麗になってきた。しかしレーザープリンタに比べればまだまだである。レーザープリンタの解像度は，おおよそ24ドットプリンタの2倍ある。したがって，印刷の品質にも相当な差がある。その差は，見ていて痛々しくなるほどである。僕はTeXを使うようになってからこのことを痛感した。CANVAS PRO-68Kは仮にもポップアートツールである。最高の品質の出力が求められる。でなければ，デザイナーには使えない。すくなくともドロー系のツールは，レーザープリンタ程度の出力ができるようになるとさらにおいしく使えるのに，と思うわけだ。

確かに，レーザープリンタは誰もが持てるものではない。最近はかなり安くなったが，まだまだ手は出ない。それは認める。シャープだってまだX68000対応のレーザープリンタなど出してはいない。しかし，それでも対応していて損はないのではあるまいか。また，PostScript*1などのスクリプト言語に対応した形式での出力など，できると嬉しいのだが……。

それから画像ファイルへの出力がない。CANVAS PRO-68Kの目的は印刷にある。だから画像を出力する必要がないというのはわかる。それはそれでいい。しかし，これはあくまで個人的な意見だが，ドロー系ツールで作られた部品は紙の上だけでなくいろいろな方面に出ていっていいものだと思う。印刷コマンドのほかにフルカラーの画像ファイルを作るコマンドもあると，新しい使い道が見つかりそうなものだ。たとえばなにかのタイトル画面に使おうと思ったら，イメージファイルの形で持っておけるととても便利だと思う。

総じて，CANVAS PRO-68Kは自己完結の傾向が強い。外部とデータをやりとりすることに消極的であるように思われる。ひととおりは用意されているのだが，本当にひととおりでしかない。

夢の(でもないけど)輪郭抽出

スキャナから画像をペイントセルに取り込む。その画像から輪郭抽出してドローオブジェクトを生成する機能はぜひとも欲しかったところ。いまのところは，取り込んだ画像はペイントセルに張り付いているだけ。それ以上の発展もなにもない。

これをドローセルに持っていけるととてもありがたい。そうなれば，手書きの下絵をスキャナで取り込んでドローセルに移し，ドローツールならではの編集機能を存分に揮うことができる。色を変えたり，拡大縮小変形したり，思いのまま。正直にいって，ドロー系のツールは，いったん部品を作ってしまえば使いでがあるのだが，部品を作るまでの段階がペイント系に比べて遥かに苦しい。たとえば線1本引くにしても，その手間は数倍である。

また，困ったことに，絵が複雑になってくると，CANVAS PRO-68Kの編集作業はどんどん重くなってくる。部品の数が増えてくるのだから当然なのだが，これまたペイント系に比べてポイントの低いところではある。この苦行(失礼)をある程度軽減してくれそうなのが輪郭抽出である。決して不可能な処理ではない。事実，世の中には取り込み画像から輪郭抽出してアウトラインデータを自動生成するソフトウェアもある。

僕の欲しいドローツール

CANVAS PRO-68Kはポップアートツールだが，僕が望むのは，CANVAS PRO-68Kが目指すところとは少し違っていて，とにかく軽い作図ツールである。仕事柄，簡単に立ち上げられて簡単に作図のできるツールが欲しいのだ。たとえば，MicroEMACSで原稿を書いていて，ちょっと図を描く必要が出てきたときにそれを痛感する。こんなときは，マウスプレーンで動作して簡単に呼び出せるようなものが嬉しい。一般的には，レポートや論文で使用する図版，企画書などで必要なプレゼンテーション用のビジュアルなどに利用できるといえばいいだろうか。結局，Mac Drawみたいなものになったりして。

カラーなどいらない，白黒で十分。カラー印刷はなにかとコストのかかるもの。ポップアートツールとしてのCANVAS PRO-68Kは優れている。機能も多い。しかし，それよりも，おもいっきり軽い常駐プログラムのほうが使える場合もあるのだ。色数と機能の多さの分だけ重くなっているのがなんとも惜しまれる。

それほど豪華なものはいらないのだが。

賢い付き合い方

CANVAS PRO-68Kにはたくさんの編集モードがある。編集にあたっては，これらのモードをうまく切り替えながら作業

するのが賢い。それを忘れると，とんでもなく重たいものになってしまう。たとえば，ある程度作業が進むまでは他のセルを表示しないで編集する，とかいったことを知っていると，作業がかなり楽になる。本文では，そうしたことはわかっているものと思って話を進めている。どうせ実際に使ってみないとわからないことでもあるし。

やっぱり資源を食うのだ

メモリが1MバイトのX68000でCANVAS PRO-68Kを動かすのはちょっとつらい。ASK68Kも組み込めないかもしれない。ついでにいうなら，アウトラインフォントはファイルで持っているので，ハードディスクの使用を勧める。RAMディスクならもっといいが。特にアウトラインフォントをフロッピーで使うのはかなりきついと思う。文字を扱わないつもりであればいいが，グラフィックツールがメモリを喰うのは常識なので，拡張はしておくべきだろう。

結論

CANVAS PRO-68Kは，ポップアートツールというだけあって，楽しいイラストレーションやポストカードなどを作るのには向いている。基本的にマウスがあればほとんどの操作はできる。いろいろな文字が使えるし，印刷もそこそこ美しい。

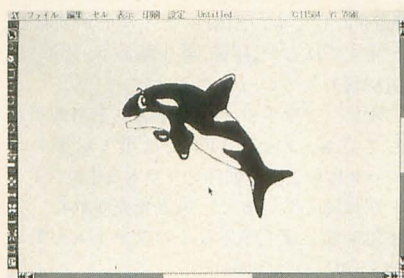
本職のデザイナーには周辺機器との入出力関係で弱い点もある。また，ビジネス文書（それも固めの）で使おうという人にはちょっと向いていない。個人でイラストレーションを描く人や，綺麗なハガキや案内状をいろいろ作る人にCANVAS PRO-68Kはお勧めである，と結論しておこう。

＊1　PostScriptは，レーザープリンタなどを制御する言語のひとつ。フォントの設定はもちろん，直線や曲線の描画などもサポートしている。スタック型言語で，再帰呼び出しも可能である。C曲線などといったフラクタル図形も数行のPostScriptのプログラムで描くことができる。NeXTなどでは，画面表示にもDisplay PostScriptが採用されている。

ドローグラフィックデータ集

CANVAS PRO-68Kの編集速度はほめられたものではない。しかも，ドローオブジェクトを構築していって好きな形を作るのはしんどい作業である。よほど熟練しないとCANVAS PRO-68Kを使いこなせるようにはならないだろう。しかし，そこはちゃんと解決策が用意されている。なにもユーザーがすべて作る必要なんてないのである。

CANVAS PRO-68Kは部品を扱うツールなのであるから，その部品をたくさん用意しておけば，それほど手間をかけなくても絵は描けそうだということになる。標準でついてくるデータディスクには干支など，いくつかのドローオブジェクトが入っていて，それらを呼び出せば年賀状のワンポイントくらいすぐ作れるわけだ。

CANVAS PRO-68Kで読み込んですぐ使えるデータ集の存在はありがたい。面倒な入力作業は一切なし。もちろん，たくさんあるドローオブジェクトの中から好きなのを選んで組み合わせることもできる。レイアウトを決めるのはユーザーなのだ。あとから文字を入れることだって簡単。

標準でついてくるデータだけではどうしてもいろいろな用途には対応しきれないということで，CANVAS PRO-68Kで使えるドローグラフィックライブラリも用意された(別売)。すでに発売されている。このグラフィックライブラリは2巻構成になっている。第1巻が個人向けのデータ集，第2巻がビジネス向けのデータ集ということになっている。定価はいずれも8,800円(税別)。

収録されている絵柄は比較的真面目なものが多い。本当に「イラストレーション」といった感じの絵柄である。まあ，変にウケを狙ったものだと使い回しがきかないだろう。

（年賀状）このデータ集には，時候のあいさつに使うものがいくつか入っている。毛筆体の「賀正」もそのひとつ。干支を入れてしまえば立派な年賀状だ。もっと凝った台詞を入れたいときは，テキストセルで作って入れれば完璧だ。

（暑中見舞い）暑中見舞いの決まり文句も毛筆体で入っていた。あとはこれに夏の風物をみつくろってやれば暑中見舞いのできあがり。

（その他）クリスマス関係のもあるが，もう過ぎてしまったな。そうそう，バレンタインデーのもあった。これはまだ間に合うぞ（いったい誰が使うんだ……）。

印刷例　CZ-8PC3使用（原寸）

POP

st. Valentine's day

フォント
ADDRESS

1234567890

ABCDEFGHIJK
LMNOPQRSTU
VWXYZ
abcdefghijklmnopqrstuvwxyz

Z'sTRIPHONY DIGITAL CRAFTデータ集

ポリゴンデータ「3D倶楽部」

Tan Akihiko 丹 明彦

3D倶楽部は，1年ほど前に紹介した，ツァイトの3Dポリゴンモデリング&レンダリングシステムである「Z'sTRIPHONY DIGITAL CRAFT」のデータ集である。制作の大変な3Dデータが多数用意されていて，ユーザーの負担を減らすようになっている。

3D倶楽部は「リビング編」と「ダイニングキッチン編」の2つがすでに発売されていて，以後，「地図編」「プライベートルーム編」と続く。

「リビング編」と「ダイニングキッチン編」は，その名のとおり，多くの種類の家具の3Dデータを用意してある。部屋のデータも用意してあるので，Z'sTRIPHONYを起動して部屋のデータを呼び出し，続いて家具のデータを次々に呼び出して部屋の中に配置するだけで，すぐにでもインテリアデザインに利用することができるようになっている。適当に構図を決めてレンダリングすれば部屋の完成予想図が手軽に作れる。

CGの生命はデータにあり

コンピュータグラフィックの作品の出来栄えを決めるものは何であるかご存じだろうか。高度なレンダリングアルゴリズム？ そうではない。

それはデータである。少なくとも僕はそう理解している。データというのは，いうまでもないが，物体の形状であり，ポリゴンの色や属性であり，質感を表すパラメータであり，マッピングであり，時には視点の座標にまで及ぶ。ユーザーがいじれる範囲のものすべてが，作品の成否を決める鍵を握っているのだ。

むろん，描画アルゴリズムも重要な役割を演じているのだが，最終的にはデザイナーのセンスにかかっている。リアルなレンダリングを可能にしたというブたい文句でCGシステムを発表しても，そのサンプルが貧弱なものだったら，誰が使ってみたいと思うだろう。ユーザーにアピールするためには，そのへんのことが実は重要なのだ。これは3Dコンピュータグラフィックだけでなく，2Dグラフィックツールやその他のさまざまなツールについて当てはまることだと思う。サンプルを充実させるのは成功するための重要なファクターのひとつではあるまいか。

ツァイトはデータライブラリの重要性をよく承知しているようだ（この3D倶楽部も，そうした思想の表れと見ることができよう)。

まず扱えるデータはいろいろな形式でファイルに保存しておける。たとえばZ'sSTAFFにしても，ZIMファイルの形式はひと通りではない。圧縮/非圧縮あり，矩形/自由形状ありといったぐあいだ。

次に，アウトラインフォントを多数用意している。Z'sSTAFFには第1水準のゴシック体および明朝体のアウトラインフォントが付属しているし，最近発売された「書体倶楽部」は毛筆体などのフォントを集めたものだ。

そして，これが一番重要な点なのだが，これらのデータは，ソフトを問わず，いやそれどころか機種すらも問わず，自由に利用できるのだ。ZIMファイルはZ'sTRIPHONYやZ'sWORD JGで取り扱えるし，アウトラインフォントはZ'sSTAFFやZ'sWORD JG，そしてZ'sTRIPHONYのどれでも使える。そして，PC-9801シリーズでも同様に使えるのである。そのために使い勝手が多少落ちている嫌いはあるものの，この方針は立派である。

データは使い回しのきくような形式にしておき，できる限り共有化する。当たり前のことのようでありながら，それすらも守れていないソフトウェアのどれほど多いことか。

ユーザー待望のデータ集

しかし，非常に重要な要素であるコンピュータグラフィックのデータ作り（モデリング）は，非常に大変な作業であるのもまた事実。人は3Dモデラーを前にして，必ず一度は石になる。パッケージに印刷されている豪華なサンプルと，自分の作ったガラクタとの間のギャップに溜息をつく。だがそれでいいのだ。誰でもそこから始めるのだ。そして，いつかはモデラーを手足のように操り，かつての目標であったパッケージのサンプルを超える作品を作るのである。

今回紹介する3D倶楽部は，まだ溜息レベルのユーザーにとって福音といえるものである。もちろん使いこなしを究めたユーザーにとってもおいしい。ともすれば複雑になりがちな3次元のポリゴンのデータを制作する手間を省き，またそこまでZ'sTRIPHONYの操作に習熟する余裕のないユーザーにも気軽に使えるデータ集である。

*

机や椅子を始めとして，ひと通りの家具は揃っている。種類も豊富だ。

この春進学や就職のために上京してくる人，フジテレビの恋愛ドラマによく出てくるようなワンルームマンションの生活に憧れている人，そんな人はこの3D倶楽部でおしゃれな都会生活の設計でもなさってはいかがだろう。……と，こういうと茶化しているように聞こえるかもしれない。実はそのとおりである。レイアウトに悩むほど広い部屋に住める人なんてそう多くはないだろう。

むしろ，もっと大人のユーザーに勧めるべきかもしれない。結婚して新居を構えるという幸せもの，また家を新築した人，引っ越す人，単に部屋の模様がえをする人でもいい。実際に家具を配置する前に，まずZ'sTRIPHONYと3D倶楽部でテストする，という使い方ができる。

しかし大事なものが欠けている。X68000だ。僕のような人間は，部屋を設計するのにまずX68000を中心に考える。快適な作業場を作るための配置は，そのあとからついてくる。どうせならX68000のある部屋を設計したいものだ。というわけでX68000の3Dデータもよろしく。

●3D倶楽部　各9,800円（税別）
ツァイト　☎03(3299)0461

解説レポート

Z'sSTAFF支援ツールZ's-EX

Tan Akihiko 丹　明彦

1月号付録ディスクのZ's－EXは，裏画面のサポート，3次元変形マッピングなど，さまざまな拡張機能を実現します。あなたのZ'sSTAFFはもうパワーアップされたでしょうか。今回はそこで盛り込まれた機能の解説を行いましょう。

先月号のディスクで配布したZ'sSTAFF支援ツールであるZ's-EXの，重い重い技術解説をお届けする。確かにプログラムも重いが,解説はもっと重いのだ。お覚悟を。

基礎知識編

Z's-EXは,Z'sSTAFF PRO-68K(以下Z'sSTAFF）の作業に割り込んで，便利な機能を使うようになっている。実は，これはとても楽で，ずるくて，しかし極めて有用な手口である。なぜなら，基礎となるグラフィックツールを作る必要はないからである。1から作らずに好きな機能を追加できるというわけである。Z'sSTAFFの慣れた環境から離れずにすむというのもメリットだ。

Z'sSTAFFを拡張するうえで，Z'sSTAFFの動作にはいくつか押さえておかねばならぬことがある。Z'sSTAFFは，アプリケーションとしてはかなり資源を贅沢に使っているほうで，およそ音源関係以外のあらゆる資源を使っているようである。テキスト画面までしっかりと徴用していて，printf( )すら使えない。

以下で述べていることは，今回のプログラムを作る上の下敷きにさせていただいたPICFILER（電脳倶楽部掲載）から大部分の情報を得ている。

待避画面

Z'sSTAFFは，ご存じのとおりウィンドウが山ほど開くシステムである。どのくらい開くかというと，スクリーンを覆いつくしてもまだ開けるくらい開く。

さて，ウィンドウシステムで頭の痛い問題は，ウィンドウを閉じたとき，または移動したときに，そのウィンドウの下にあったはずの画面をどうやって復活させるか，ということであろう。ウィンドウを消したらその跡は真っ黒だった，というのではウィンドウシステムの意味がない。

まず思いつくのが，ウィンドウ1枚を開くごとに，その下にあった画面を記憶するという方法だ。ところがこれはだめ。ウィンドウが1枚しか開けないシステムならそれでもよいが（PC－9801用のZ'sSTAFFでは確かウィンドウは1枚だったが……)，同時に2枚以上なら破綻する可能性がある。次のシチュエーションを想像してみてもらいたい。

1)　ウィンドウ甲を開く。ウィンドウ甲の待避領域にはその部分のG－RAMの内容が入る。

2)　ウィンドウ乙を開き，ウィンドウ甲と一部が重なるように移動する。ウィンドウ乙の待避領域にはその部分のG－RAMの内容，すなわちそこにあった画面とウィンドウ甲の一部が入る。

3)　ウィンドウ乙を上にする。

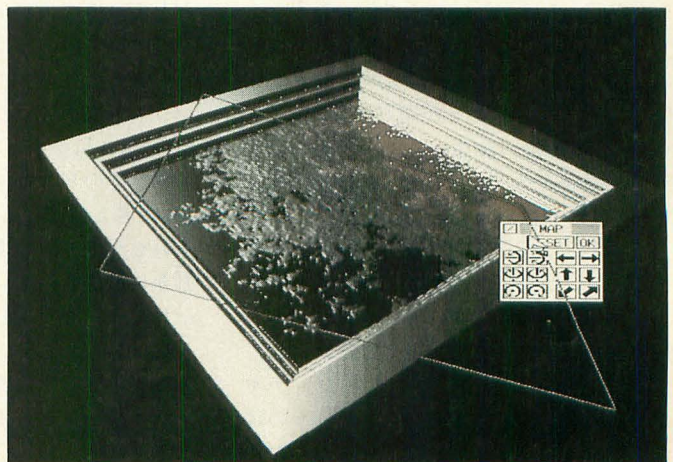
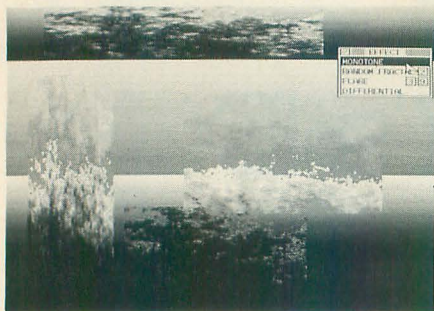
4)　ウィンドウ甲を閉じる。ウィンドウ甲が消えた跡には，もとの画面を待避領域から読みだしてG－RAMに書き戻しておくが，ウィンドウ乙のほうが上にあるのだから，ウィンドウ乙と引っ掛かる部分にまで書いてはいけない。

5)　ウィンドウ乙を閉じる。待避領域には，ウィンドウ甲の一部ががが入っていたため，それまでG－RAMに書き戻されてしまい，画面にはウィンドウ甲の一部がごみとなって残されることになる。

もちろん，もう少し真面目に処理すればこのようなことは起こらないのであるが，ただでさえ重いウィンドウシステムの処理がもっともっと重くなる。2枚ならまだそれでもいいが，これが何十枚ともなると想像するだけでも恐ろしい。

それに，こちらのほうがもっと深刻なのだが，画面を覆いつくしてもまだ余るほどのウィンドウ面積分の待避領域がG－RAMの容量を超えてしまうのは当然のことなのである。だから，いっそのこと待避画面をどこかに1画面分気前よく取ってしまうのがお徳用なのだ。そうすれば，ウィンドウを閉じたときの書き戻しはその待避領域から正直に読んでくればすむ。そのウィンドウと重なっているウィンドウ（こういうのをオーバーラップしているという）を書き直すだけの手間ですむ。

この待避領域には，G－RAMと同じ512Kバイトが必要である。しかし，そんな大きさの配列を取るのは，当時（1987年）のユーザー状況を見れば犯罪に等しかったと思う。メインメモリが1Mバイトが圧倒的多数派だった頃のことである。

そこで，どのX68000も持っている広大な領域，G－RAMと同じ512Kバイトの領域，すなわちテキストVRAMがまるごと待避領域に当てられた。線を引いたりペイントしたりといった動作は，この待避領域とG－RAM（当然ウィンドウが表示されていない部分のみ）の両方に対して加えられる。また，両方に描画しないと本当に描

裏画面を3D変形してマッピング

RANDOM FRACTAL

DIFFERENTIAL

いたことにはならない。たとえばG-RAMだけに描いたとしたら，自分では描いたつもりでも，ウィンドウを動かしたり消したりしたときにボロが出る。もうひとつ，細かい話だが，テキストVRAMは，単なる配列として使う（当然表示もしない）ので，テキストVRAMの同時アクセスモードをキャンセルしておくこと。拡張プログラムを作る際には，この2つのことに注意されたい。

こうしてくれたおかげで，今回のZ's-EXのウィンドウも待避領域を取ることなく実現できた。これがメインメモリの配列だったら問題は難しくなっていたことだろう。テキストVRAMのアドレスは決まっているが，配列のアドレスはわからないからだ。

蛇足ながら付け加えておく。はじめからテキストVRAMにウィンドウを開けば，そんな巨大な待避領域なんていらなくなる，と思った人は，非常に鋭い。確かに原理的にはそのとおり。テキストにウィンドウを開けば，どこで開こうが閉じようが，G-RAMは一切傷つかない。ウィンドウどうしがオーバーラップしている場合の処理だけ考えておけばいいことになる。

しかし，ことZ'sSTAFFの場合はそううまくはいかない。つまり，パレットウィンドウはどうするの？　というわけ。あの極彩色のウィンドウを表現するためには，どうしてもG-RAMが必要なのだ。テキストVRAMを待避領域に使うというのは，さんざん悩んだうえでの選択だったのだろう。ま，どちらにしてもテキストVRAM

FLARE

をウィンドウ表示に使うか待避領域に使うか，の差で使用する資源に変わりはない。

マスキング

Z'sSTAFFは，画面の任意の領域にマスクをかけてその下の画面を保護することができる。マスク領域には一切の描画ができない。

Z'sSTAFFをお使いの方はご存じであろうが，マスクをかけた領域は青くゆっくりと点滅している。これは割り込みを使って細かくパレットを変えることで実現している。

ここで問題になるのは，いったいどの色のパレットを変えているのかということ。

Z'sSTAFFを起動するときにスペースキーを押していると，G-RAMを保存したまま起動するのだが，その際輝度ビット（1ピクセルは16ビット＝0～65535で表現されるが，その最下位ビットが輝度ビットに割り当てられている）が立っているピクセルにマスクをかけるようになっている。

しかし，輝度ビットの立っているピクセルだけの色を変えるのは難しい。単純計算でも32768色あるのだ。ところで，X68000は65536色モードでもパレットは変えられるが，それは制限つきで，ひとつのパレットを変えると同時に256色が変わってしまうのだ。このへんはややこしい。

代表として，パレットコード$0001のパレットを変えている。しかし，そのためにはG-RAMの中で輝度ビットの立っているすべてのピクセルのパレットコードを$0001に変えねばならない。しかしうかつにそれをやると，今度はマスクを取り去ったときに保護されていた絵が出てくれないということにもなる。ちなみに，パレットコード$0001とは，黒の輝度ビットを立てたのと同じこと。その輝度ビットを取り去れば，マスク部分は真っ黒になってしまうことになる。

そこで，またテキストVRAMの出番である。テキストVRAMのほうでは，マスクに対応する部分に輝度ビットを立てるだけで，マスク以外の部分には手をつけない。マスクを取り去るときは，テキストVRAMから読み出して書き戻せばよい。

マウスカーソル

待避画面とマスキングからわかるように，Z'sSTAFFではG-RAMは表示にのみ使われていて，実際の作業はテキストVRAMで行われていると考えるのが無難である。

となると，玉突き式に新たな問題が浮上する。マウスカーソルである。

通常のシステムでは，マウスカーソルはテキストVRAMを使って表示している。ところがそのテキストVRAMは，Z'sSTAFFでは待避画面に使われてしまっている。1バイトたりとも余っていない。そこで今度はスプライトが浮上してくることになる。

Z'sSTAFFを使ったあとに適当なスプライトエディタを立ち上げれば，スプライトに定義されているパターンがわかる。マウスカーソルはパターンコード0のスプライトに定義されている。マウスカーソルの表示ルーチンをフックして，テキストVRAMにカーソルを書き込むかわりにスプライトを表示しているのだろう。Z'sSTAFFが768×512ドットモードでは動作しない(CRTコントローラに悪いことをすれば，768×512ドットモードでも65536色の表示はできる）というのは，実はこのスプライトがネックになっているからである。

またZ'sSTAFFでは，なにか時間のかかる作業をしているときはマウスカーソルが腕時計の形になる。それもパターンコード0のスプライトのパターンを書き換えているだけ。

ところで，いくつかあるメニューのうち，G-RAMを使っていないものがひとつある。画像取り込みがそれ。このメニューだけは，マウスカーソルと同様にスプライトを使っているようだ。イメージユニットを使うだけに，ウィンドウを避けて取り込むということができないせいであろう。

キー入力

メニューバー（「ファイル」「パレット」「ペン」……と書いてある細長い棒）を消したり出したりするのにはスペースキーを使う。このスペースキーの読み取りにはIOCSコールのB_BITSNS (IOCSコール

番号$04）を使っている。Z'sSTAFFはこのIOCSコールを頻繁に行っている。

そこで，あるキーを押したときに拡張ルーチンが起動するように設定したい場合は，このIOCSコールのベクタを書き換えるという方法でキー入力を奪い取ることができる。むろん，その起動キーでなかった場合は，Z'sSTAFFに戻り，そのキー入力を伝えてやる必要がある。そうでないと，スペースキーが効かなくなってしまうからである。

Z'sSTAFFでキー入力をする場面はいくつかある。たとえば文字入力がそのひとつだが，日本語FEPのASK68Kは幸いにしてB_BITSNSを呼んではいないようなので，心配することはない。

そこでZ's-EXの制作

以上のことを念頭に置きつつ，Z's-EXを制作した。以下は各プログラムの解説である。プログラムリストは大きすぎてちょっと誌面には載せられないが，すべて先月号の付録ディスクに（DISK2）に収録されているので各自参照してほしい。

PICFILERは常駐プログラムであったが，Z's-EXはそうしなかった。Z'sSTAFFをチャイルドプロセスとして呼び出す，ごく普通の実行ファイルである。どうしてこうなったかというと，ひとつは常駐解除を忘れたときに思わぬトラブルの種になるということ。もうひとつは，常駐プログラムを書くのはそれなりに面倒だったからである。

Z's-EXは，Z'sSTAFFの待避画面とは別にもう1枚の裏画面を取る。512Kバイトの配列をメインメモリに取るなどという，なかなか凶悪なことをしている。このためメインメモリ1MバイトのX68000での実行はほぼ絶望的である（とはいっても，もともと1MバイトでZ'sSTAFFを動かすのはかなりきついものだった）。

ウィンドウマネージャもどき
——window.c——

Z's-EXは，同時に1枚のウィンドウしか開かないが，とりあえずウィンドウシステムである。

「使い回す可能性があるシステムのコーディングは1回はやり直したほうがいい」という鉄則はやはり真実だったようで，作ったあとで振り返ると，ここはこうすればよかった，あそこはああすればよかった，といった後悔が渦巻いている。ウィンドウマネージャを作った段階では完璧だと思っていたが，そのあとを作っているうちにあちらこちらと矛盾が出て，そのたびに対症療法を繰り返した。その結果，少々汚いプログラムになっているのだが，ちょっと見にはそれほど汚くはない。

さて，各ウィンドウの上にはいくつかのアイテムが並んでいる。それをマウスで指定する。というだけなら簡単だが，操作性や見た目のよさを重視するならば，ことはそう単純にはいかない。

・マウスカーソルをアイテムの上に持ってきたとき，表示が反転するアイテムとしないアイテムがある。反転しないアイテムの代表は，タイトルバーである。ここをつかんでドラッグするとウィンドウを動かせる。いちいち反転するとうっとうしい。

・マウスボタンを押したら即時呼び出し元に復帰するべきアイテムとボタンを離すまで待つべきアイテムがある。即時復帰するアイテムの代表は，PICFILERのファイルウィンドウ。左ボタンまたは右ボタンでスクロールするのだが，これは押しっぱなしにしたときには連続してスクロールするのが望ましい。

以上のようなことを考え合わせて，ウィンドウのアイテム管理情報は図1のようにしてある。プログラムリストの解読をする際には，参考になることだろう。

この情報は，新しいウィンドウを開くたびに更新される。グローバル変数にいちいち代入しなおすなどという間抜けなことをしている。まとめて構造体で持っておかなかったのは失敗だったのだが，今となっては手遅れ。

ともかく，ウィンドウマネージャもどきは，この情報を参照しながら，マウスカーソルの座標を読み取り，必要ならアイテムを反転させて，マウスボタンを押されたらそれなりの処置をとる。そして，押された

図1　ウィンドウマネージャのアイテム管理情報

```
typedef ITEM short[10];      …… ひとつのアイテムにつき10ワードの属性を要する
int   win__x0, win__y0;      …… ウィンドウ左上の座標
int   win__x, win__y;        …… ウィンドウのサイズ
int   win__n;                …… アイテムの数
ITEM  win__i[win__n];        …… 各アイテムの情報

    i 番目のアイテムの情報の内訳（0 ≦ i ≦ win__n-1）

win__i[i][0]
        最下位ビットが1（つまり奇数）なら,
        このアイテムの上にマウスカーソルがあるときに表示が反転する。
        特に，この値が255のときはウィンドウクローズ,
        254のときはウィンドウ移動としておく（特に意味はない）。

win__i[i][1]
        この値が0のときはマウスのボタンが押されたら即時復帰する。
        1のときはボタンを離すまで待つ。

(win__i[i][2], win__i[i][3])−(win__i[i][4], win__i[i][5])
        マウスカーソルがこのレクタングルで示される領域の中にあれば,
        このアイテムが選択されているとみなす。

(win__i[i][6], win__i[i][7])−(win__i[i][8], win__i[i][9])
        アイテムが選択されているとき，表示を反転させる領域を示すレクタングル。
        win__i[i][0]が偶数（反転しない）の場合は意味を持たない。
```

謹賀新年PRO-68KのDISK2に収録されているZs-EXの内容

ボタンの状態とアイテムの番号を戻り値として呼び出し元のルーチンに返す。

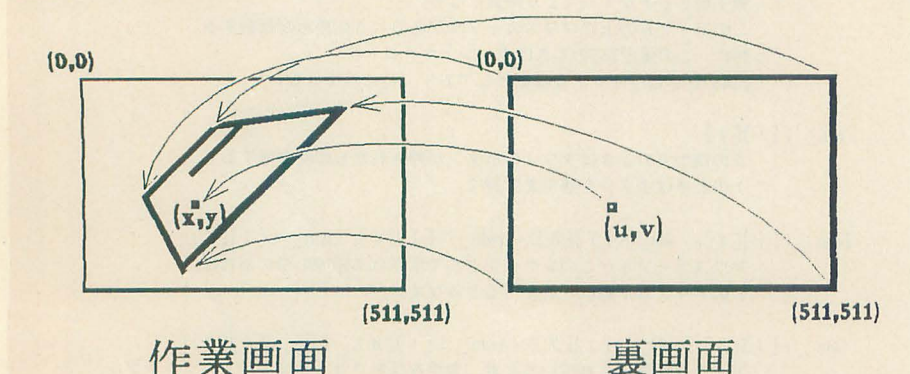
ウィンドウの移動については専用のサブルーチンを持っている。動かす前のウィンドウをテキストVRAMの内容で塗り潰して，新しい位置にウィンドウを表示しなおすという方法は間抜けなので，できる限りブロック転送を使っている（この部分はアセンブラ)。

ウィンドウのクローズはアセンブラで書いた転送ルーチンを呼んでいるだけ。

ほかに，ちょっと重大そうな処理に関しては実行する前に「本当にいいか」と確認を求めさせたい場合も出てくる（うっとうしいから不要だという意見もある。確かにいえてる)。そのサブルーチンもウィンドウ関係に含めている。実は，この確認ウィンドウもウィンドウマネージャで処理しているのだ。

文字表示は12ドットとしている。シングルウィンドウなのだから文字を小さくしてもたいしてメリットはないのだが，ウィンドウは小さくまとまっているのが好きなもので，あえて16ドットを使わなかった。ここではBASICライブラリのsymbol( )関数を使うという真似をしている。だから実は何ドットの文字でも手間はたいして変わらない。テキスト画面が使えないという時点で，そもそもprintf( )に逃げることが不可能になっていたのだから。

PICFILER
——picfiler.c——

プログラムは長めだが，たいしたことはしていない。picファイルのロード/セーブにしても，オリジナルのxpic.sのルーチンを呼んでいるだけである。ただし，xpic.sのうち,スーパーバイザモードに入る/スーパーバイザモードから抜けるという部分は取り除いてある。これは，Z's-EXがすでにスーパーバイザモードで走っているからである。

ほかに気をつかったところといえば，ファイル名をキーボードから入力させるのに，やはりscanf( )も使えないので，自前で文字列入力ルーチンを作ったことと，拡張子を強制的に“.pic”にしたことくらいである。ファイルウィンドウにしても，ディレクトリと“*.pic”のファイルしか表示しない。これはことによると迷惑だったかもしれない。

マスキングのロード/セーブに関しては，まだサポートしていない。

ALTERNATE SCREEN
——zs.c——

裏画面と表画面の切り替えを行う。処理の中身は単純な転送である。つまり，裏画面の配列の中身と待避画面（テキストVRAM）の中身を入れ替える。G-RAMにも一緒に転送している。ただし，マスキング部分（輝度ビットが立っている部分）については，G-RAMにはそのままでは転送しないで，パレットコードを$0001に強制的に変更してから転送している。

MAP
——map.c,transfer.s——

裏画面を1枚の長方形に見立て，表画面に任意の位置・角度で張りつける。Z'sSTAFFには矩形領域の自由変形というのがあるが，これはそれとは少々異なる。あくまで3次元空間内に裏画面を置いたと考えているので，裏画面には遠近感がついているはずである。

この処理は，3次元コンピュータグラフィックでいえばテクスチャマッピングに相当する。ただ，アルゴリズムは多少異なる。一般のマッピングには透視逆変換が必要なのだが，それと同じことを，別の方法で実現している。

長方形の枠(反転したラインで表示する)を移動/回転させ（マウスで押したアイコンに対応した動きをさせる)，位置を決めたら本処理に入る。

処理はややこしいので，粗筋をまず述べる。

1)　まずソリッドスキャンコンバージョンで，画面上の長方形の中にあるピクセルをピックアップする。そのピクセルの座標をp (x,y)とする（ピクセル(x,y)にマスキングがかかっていたなら，処理をスキップし，次のピクセルの処理に移る)。

2)　次に，ピクセル(x,y)が裏画面のどのピクセルに対応するかを求める。この座標を(u,v)とする（ピクセル(u,v)にマスキングがかかっていた場合も，処理をスキップする)。

3)　ピクセル(u,v)の色をピクセル(x,y)に書き込む。

これを繰り返すだけなのだが，このなかでどこが面倒なのかというと，なんといっても2)の座標変換の部分である（図2)。

もともとは長方形であった裏画面は，作業画面のなかでは変形している。作業画面という「窓」を通して見える3次元空間を考える。そこに裏画面を浮かべているのだ，と思うことにしよう。

座標変換処理の第1段階として，ピクセルに対応する3次元空間内での位置を知る。これは，裏画面の長方形を含む平面の方程式と，視点から窓の中の一点（ピクセル）へ伸ばした直線の交点を求めればよい。視点は，Z's-EXを使ううえでは特に意識する必要はないが，図3のように固定してある。視点を決めれば，あとはレイトレーシングと同じ要領で視線を発生させ，平面との交点を求める。こうして求めた平面上の座標を(xi, yi, zi)としよう。

第2段階として，上で求めた座標を裏画面の実際の座標系に変換する。裏画面の位置と角度を決定した時点で，裏画面の四隅の（3次元空間内の）座標はわかっている。したがって，(xi, yi, zi）が長方形の中のどのあたりにあるかは，簡単な内分比の計算で求めることができる。

以上のことを図4にまとめてある。ただし図4の式はあくまで原理的なもので，実際にCでコーディングするときはもっと効率的にするためにいろいろと姑息なことをしている。ただ，一番大事なテクニックであるはずの整数化を怠っていたりする（反

図2　マッピング座標系

省)。

なお，裏画面の位置はいつでも初期化できる。初期化すると，表の作業画面と裏画面がぴったりと重なる格好になる。このときに限り，遅い座標変換を使わずに，直接転送するようにしている。ここはアセンブラで書いてあるので，そこそこの速度は出ている。これは画面合成に多用している。

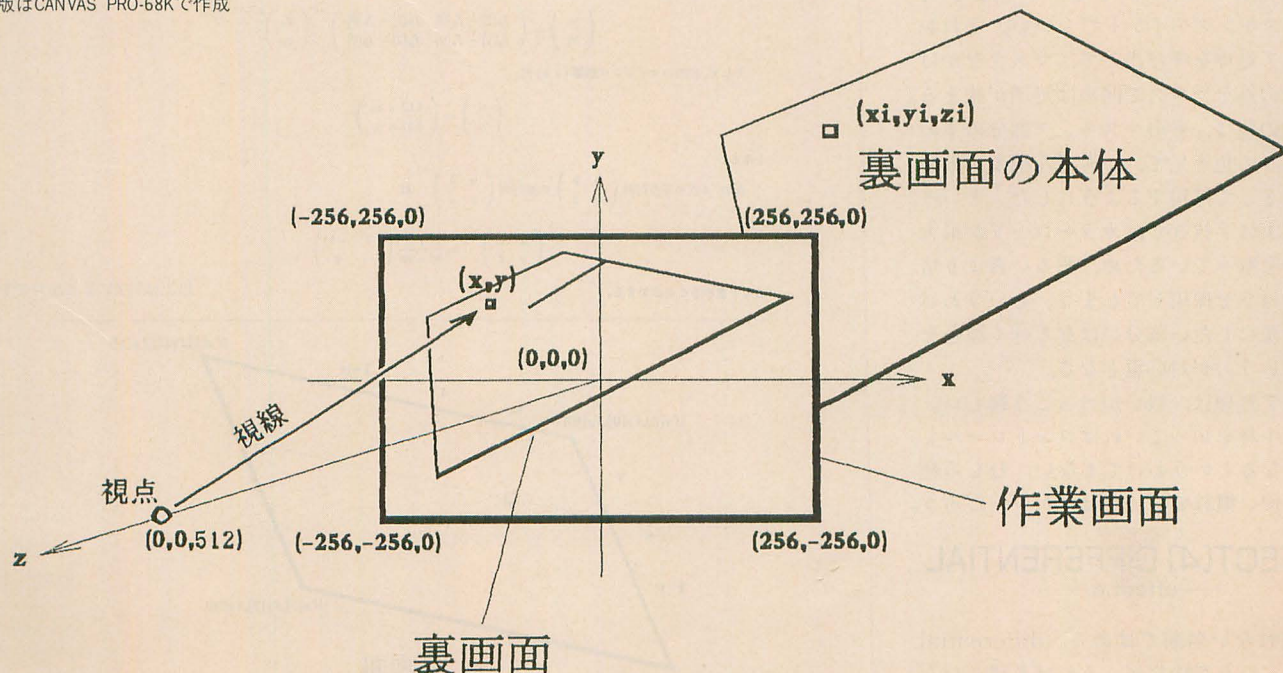
MASK PAINT
——zs.c,rev.s——

これは素晴らしい手抜きをしている。IOCSコールのペイントルーチンを使ってカラーコード$0001でペイントしている。ただ，これだけではテキストVRAM上の待避画面にマスク情報が伝わらないので，メインメニューに戻るときにG-RAM全体をサーチし，カラーコードが$0001のピクセルに対応する待避画面のピクセルの輝度ビットを立てるようにしている。マスキングペイントからメインメニューに戻るときに1～2秒間マウスカーソルが時計マークになるのはこのためである。

EFFECT(1) MONOTONE
——effect.c,rev.s——

ピクセルの色成分をR,G,Bとしたとき，そのピクセルの輝度は，

M=(0.11×B+0.30×R+0.59×G)

で与えられる。青，赤，緑の色成分は等価ではない。目に自然な輝度を求めるためには単なる平均を取ってもだめで，この式のように重みをつける必要がある。実数を使うのはいやだったので，

M=(28×B+77×R+151×G)/256

と，8ビットのゲタをはかせて計算している。これにより，輝度Mに0から31の値が得られる。それをいちいちシフトしてグレースケールのカラーコードを作ってもいいのだが，たかだか32個なのだから，テーブルにした。テーブルは呼び出し元（Cのプログラム側）で用意し，その先頭アドレスをアセンブラのプログラムに渡すようにしている。したがって，テーブルを作り変えれば，セピア調変換も可能である。

テーブルを用意するという考え方は，本誌1990年10月号のX68000マシン語プログラミングからもらってきた（テーブルを使うくらいのことを自分で思いつかなかったというのがそもそも問題なのだが)。

EFFECT(2) RANDOM FRACTAL
——effect.c——

本誌1990年9月号のグラフィック特集で紹介した，自然物の表現に便利なランダム・フラクタル処理も組み込んでみた。まったく同じ処理をしているが，処理を整数化して少し速くなったことと，ランダム・フラクタル格子を生成する処理を別プログラム（rfbuild.x）にして独立させたというところが違うといえば違う。

高速化したぶんにはなんの迷惑もかからないが，ランダム・フラクタル格子の初期化がZ's-EX本体から行えないというのは少々まずいかもしれない。なぜなら，毎回同じパターンが出てくるからだ。変形の強さなどのパラメータは変えられるようにしてはあるが，パターンまでは変えられない。ランダム・フラクタル格子は先月も説明したとおり，別のデータファイルrf.datになっている。したがって，パターンを変えたいときは，rfbuild.xに別のパラメータを指定してrf.datを作り直せばいいのだが，Z's-EXを抜けなくてはならないというのが間抜けである。なにぶん一瞬で実行が終わるプログラムではないので，Z's-EXの中に組み込むのをやめてしまったというのが真相ではある。

EFFECT(3):FLARE
——effect.c——

フレア処理は，画面の一部を光源に見立て，その周囲を淡く光らせる。実は，フレア処理を設計する段階で，この「画面の一部」をどうやって指定するかを決めるのに少々悩んだのだ。光源といっても，点光源ではおもしろくない。自由形状にできるというのは必要であった。結局はマスキング部分を光源と思うことにしたのであった。画面全体を処理するのは遅くていやだったので，マウスで指定した矩形領域の中のマスキング部分だけを光源にする。

あるピクセルのフレアの輝度は次のよう

図3　視点の座標系

にして求める。そのピクセルを中心とするある広さの領域をサーチし，その中でマスキングがかかっているピクセルの数を数える。次にその数を適切なフィルタに通してフレアの輝度に変換する。

サーチする広さはランダム・フラクタルと同様のパラメータになっていて，変えることが可能である。

フィルタとなる関数は，いろいろと試した結果，サーチする領域の面積をs，その中のマスク領域の面積をmとして，

$$f(m)=1-(s-3\times m)^2/s^2 \quad (m<s/3)$$
$$(m\geqq s/3)$$

と決めた。ピクセルの近くのマスキング部分がサーチする領域の3分の1を超えた段階でそのピクセルのフレアは最大になる。経験的に見て，それくらいが自然なようだ。

高速化の話。マスク領域は効率的に数えないとどうしようもなく遅くなってしまうのだ。初期バージョンでは，ピクセルの周りのマスク領域を正直に数えていた。ところが，ちょっと考えるとわかるのだが，あるピクセルの周囲のマスク領域は，そのひとつ前，つまりその左隣のピクセルのマスク領域と非常によく似ている。ほとんど同じものを何度も数えるのは間抜け以外の何物でもない。そこで，できる限り過去の結果の履歴を取っておき，数えるのは新しくサーチする部分だけにした。このため，若干ループの組み方が複雑になっている。

最後に，使ううえでの注意。まず，光源にしたい部分にマスクをかけておく。それはZ'sSTAFFのマスクでもいいし，Z's-EXのマスキングペイントでもいい。それからフレア処理を呼び出して，マスクをかけた部分の外をマウスで囲めば処理が始まる。

光源の色は，そのマスキング部分の下のピクセルの色を見て，いちばん明るいものを代表として採用するようにした。といっても，実は手抜きで，カラーコードが最大のものを取っているため，明るい青より暗い緑のほうを採用してしまう。というわけで，光源にしたい部分にはなるべく原色を使うというのが対応策となる。

フレア処理は，扱いがけっこう難しい。処理の中身を知っていればコントロールしやすくなるというわけでもない。むしろ経験的に使い慣れていったほうがいいだろう。

EFFECT[4] DIFFERENTIAL

——effect.c——

耳慣れない名前ではある。differentialは微分という意味だが，それほど難しいものではない。隣接するピクセルとの輝度の

図4 座標変換

枠の中心の座標を (c_x, c_y, c_z) とする。

枠の四隅の中心からの相対座標を $(f_x[j], f_y[j], f_z[j])$ とする。$j = 0,1,2,3$ で，それぞれが左上，右上，左下，右下隅に対応する。たとえば左上隅の座標は $(c_x + f_x[0], c_y + f_y[0], c_z + f_z[0])$ となる。

平面の法線ベクトルを (n_x, n_y, n_z) とする。

中心の座標は移動コマンドで変化し，枠の隅と法線は回転コマンドで変化する（回転コマンドは中心まわりの回転である）。

図3より，ピクセルの座標 (x, y) に対して $(0 \le x, y \le 511)$，

$$\begin{aligned} x_d &= (\text{double})(x-256)\times 1.5 \\ y_d &= (\text{double})(256-y) \end{aligned}$$

ここで x 座標を1.5倍しているのには，ディスプレイの縦横比を補正する意味がある。

視点は $(0,0,v_z)$，ただし $v_z = 512$。

平面の方程式は，

$$n_x(x-c_x)+n_y(y-c_y)+n_z(z-c_z)=0$$

となる。

視線（直線）の方程式は次のように表される。

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v_z \end{pmatrix} + t \begin{pmatrix} x_d \\ y_d \\ -v_z \end{pmatrix}$$

この2式を解いて，

$$t = \frac{n_x c_x + n_y c_y + n_z c_z - n_z v_z}{n_x x_d + n_y y_d + n_z v_z}$$

を得る。

視線と平面の交点の座標 (x_i, y_i, z_i) は，いま得た t を用いて，

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v_z \end{pmatrix} + t \begin{pmatrix} x_d \\ y_d \\ -v_z \end{pmatrix}$$

以下は，枠の左上からの相対座標で考える。つまり，枠の左上が原点になるように平行移動を行う。枠の左上の座標 $(x_{off}, y_{off}, z_{off})$ は次のように計算される。

$$\begin{pmatrix} x_{off} \\ y_{off} \\ z_{off} \end{pmatrix} = \begin{pmatrix} c_x + f_x[0] \\ c_y + f_y[0] \\ c_z + f_z[0] \end{pmatrix}$$

交点の座標 (x_i, y_i, z_i) の表式も次のように変える。

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} x_i - x_{off} \\ y_i - y_{off} \\ z_i - z_{off} \end{pmatrix}$$

枠の四隅の座標と，平面と視線との交点の座標から，マッピング座標 (u, v) を計算する。たいていの場合は，x 座標と y 座標だけから計算することができる。

(x_i, y_i) は，枠の四隅の座標 $(f_x[j], f_y[j])$ を $u_i : 1-u_i$ および $v_i : 1-v_i$ に内分している。(x_{off}, y_{off}) からの相対位置で考えると，

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = u_i \begin{pmatrix} f_x[1]-f_x[0] \\ f_y[1]-f_y[0] \end{pmatrix} + v_i \begin{pmatrix} f_x[2]-f_x[0] \\ f_y[2]-f_y[0] \end{pmatrix}$$

行列の形式に改めると，

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} f_x[1]-f_x[0] & f_x[2]-f_x[0] \\ f_y[1]-f_y[0] & f_y[2]-f_y[0] \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$

したがって，(u_i, v_i) は

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} f_x[1]-f_x[0] & f_x[2]-f_x[0] \\ f_y[1]-f_y[0] & f_y[2]-f_y[0] \end{pmatrix}^{-1} \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

そして，目的のマッピング座標 (u, v) は，

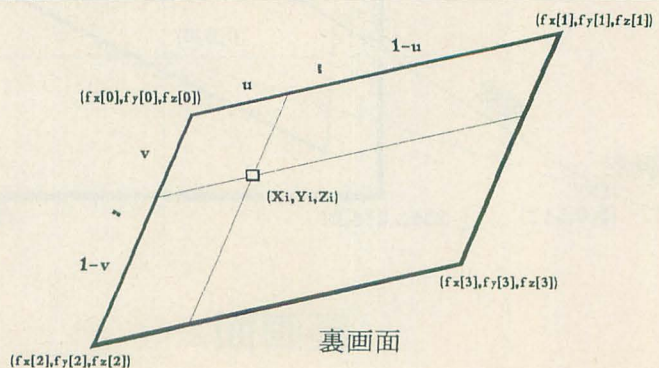
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 512 \times u_i \\ 512 \times v_i \end{pmatrix}$$

となる。

なお，2次の正方行列 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ の逆行列 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}$ は，

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

として求めることができる。

以上はTEXによる出力です

差を計算し，モノクロ32階調で表示しているにすぎない。

実行にあたっては，まず指定された領域を白黒の輝度に変換する。これは本質的に必要な処理ではないのだが，たとえば赤成分の差だけを表示するよりは，赤青緑の3成分を合わせた輝度のほうがより自然なのではないかという判断による。なによりも，特定の色成分に偏っているような絵に強いのだ。白黒化してしまえば，赤青緑のどの成分も等しい値なので，輝度の差を求めるのに，青成分の差を取ってもかまわない。

白黒化がすんだら本題の微分処理に入る。隣接するピクセルと書いたが，画像は平面，つまり2次元なので，x方向の輝度変化とy方向の輝度変化を取る。それを足したもの（本当は単純に足すことは感心しないのだが，結果はまともなのでこれで通した）をそのピクセルでの微分とするのだ。

$$D(x,y)=(M(x+1,y)-M(x,y))+(M(x,y+1)-M(x,y))$$

こうして微分D(x,y)を求めることはできるが，それを視覚化するにはもうひとひねり必要だ。なぜなら，輝度変化は増加とばかりは限らず，減少もありうるからである。画像は常に明るくなったり暗くなったりを繰り返す。すると，D(x,y)が負の値をとることも十分に予想できる。対して，輝度は常に0以上でなくてはならない。

ここで解決策は2つある。ひとつは，微分の絶対値をとること。これならつねに0以上になる。もうひとつは，微分にある定数を足したものを輝度とするというもの。ゲタをはかせて負の数をむりやり正の数にしてしまおうというわけ。今回は後者を採用した。Dが負の場合（輝度が減少傾向にある）は暗く，正の場合は明るく表示される。このほうが自然であろう。

処理自体は単純だし，そこそこ速いが，けっこう妙な効果が出ておもしろい。いってみれば，石造りのレリーフといった風情である。もし気が向いたら，Z'sSTAFFのウィンドウを閉じないでSキーを押してZ's-EXを呼び出し，ウィンドウに対してこの微分処理を施してみてもらいたい。Z'sSTAFFがたちまちにしてSX-WINDOWに変身することであろう。

開発を終わって

今回のディスク関係では，僕にしては珍しくアセンブラを多用した。アセンブラを使ってみて改めてありがたく感じるのは，68000のレジスタの多さである。たいていの処理はワークエリアを用意する必要がない。必要なパラメータはレジスタにしまっておけばいいのだ。できる限りレジスタを使うことで，高速化も図れる。

それから，デバッガには感謝しなくてはなるまい。実は，デバッガをデバッグに使ったのは今回が初めてなのだ。エラーが出るときはプログラムが悪いのだから，直ちにエディタに戻ってソースリストとにらめっこするという作法で今までやってきていた。いや，やってこれていた。

ところが，今回は正体不明のエラーが多発するのに悩まされた。僕にとってZ'sSTAFFはまったくのブラックボックスなので，そのエラーの原因がアルゴリズムにあるのかZ'sSTAFFとの相性の悪さにあるのか，自信が持てなかったのだ。いつもより早いディスクの締め切りも迫った頃，窮しても通じなかった僕は，ついにZ'sSTAFFがなにか悪いことをしているのではないかと疑ってしまったわけだ。結果的にはそれはまったくの見当違いだったのだが，それくらい僕はあせっていたのだ。

そんな折り，「デバッガを使え」という忠告を受けた。Z'sSTAFFはテキスト画面をワークエリアに使っているので，コンソールは使えない。そこで，出力はリダイレクトでプリンタに出すことにし，文字どおり手探りでのデバッグ作業となった。今思えば，RS-232Cで別のコンピュータにつないでターミナルデバッガにするという方法もあった。マシンが2台必要になるが，マシン室にはX68000が何台もある。

ブレイクポイントを設定し，レジスタの値をにらんでいるうちに，とんでもなく恥ずかしいバグを見つけた。さっそく潰したら，ちゃんと動いてくれた。

その昔，本誌で「7度デバッグして人を疑え」という祝一平氏の格言を聞いた。その頃の僕はひとりの読者にすぎなかったが，今になってその言葉を実感したのである。

それからメモリを4Mバイトに拡張しておいて本当によかった。MicroEMACSからCOMMAND.Xを呼び出して，そこでgccを走らせる。コンパイルが終わったらそこからZ's-EXを実行する。もちろん，Z's-EXはチャイルドプロセスでZ'sSTAFFを呼び出す。これだけやってもまだメモリに余裕はある。アンドゥが使えるくらい余裕がある。

しかし，このために，消費する資源が犯罪的に大きくなってしまった。少し反省。

*

Zs_EXは今後もいくつかの拡張機能を用意する予定である。ご意見や，欲しい機能などがあればお寄せいただきたい。

Z'sSTAFF PRO－68Kの魅力と問題点

とにもかくにも，Z'sSTAFFの最大の魅力は使いやすさ。これに尽きる。よく，ゲームバランスという言葉が使われるが，これはグラフィックツールでもビジネスソフトでも重要なファクターだ。いくら機能が強力であってもバランスが悪くてはうまく使いこなせない。

ところで，機能ということに関して私が日頃から意識している評価基準は，

1）どうしても不可能なものよりは，多少使いづらくとも可能なもののほうがよい。

2）可能なものどうしであれば，強力なものより使いやすいもののほうがよい。

という2点である。

ペイント系のグラフィックツールの場合，できる限り自由に美しい絵を描きたいという人にとって，使える色数は少ないものよりは多いほうがよいし，解像度も低いものより高いほうがいい。逆に最悪でも好きな場所に好きな色の点を打つ機能さえあれば，自由な絵を描くことは不可能ではない。極端な話だが，これが第一の前提だ。

Z'sSTAFFは，32768色もの色が使え，サイズも512×1024ドット。そのうえで，32768色を扱うための機能が一通り揃っている。つまり，努力すればどうしてもZ'sSTAFFで描けないものというのはほとんどないことになる。

Z'sSTAFFにはいわゆる多彩な表現といった機能がほとんどない。発表時には，ボカシの利くペンやグラデーションだけでも相当びっくりしたものだが，特殊効果に属する画像処理機能に関する弱さは当初から指摘されていた。

にもかかわらず，Z'sSTAFFを凌駕するものは現時点ではないといってよい。機能的には優れたものが今後とも出てくる可能性があるが，使い勝手の面でこれを超えるものを作るには相当に腰を据えてかからねばならないだろう。

しかし，そのZ'sSTAFFにも大きな弱点がある。入出力関係である。CANVAS PRO-68Kの場合にも指摘されていることだが，スキャナからの取り込み，そしてプリンタのサポートだ。今月号で紹介されているFine Scanner-68の場合は専用ソフトが強力なのでデータレベルの連動で十分活用できるが，シャープ純正のカラーイメージスキャナを利用する場合にはモノクロの画像取り込みで苦戦を強いられるだろう。また，プリンタも48ドットのハードコピールーチンがサポートされておらず，24ドットでのプリントアウトにしても必ずしも優秀とはいえない。

Z'sSTAFF PRO－68Kは描画面では実に素晴らしいユーザーインタフェイスをもっている。グラフィック機能に魅力を感じてX68000を購入した人は，このソフトぐらいはぜひ手に入れるべきだ。58,000円と多少値は張るが，改めてX68000の価値を実感できるはずである。機能面では要望も多いが，それらは今回のZs-EXのようなかたちでもサポートできるだろう。しかし，ツァイトには入出力の面でもう一歩がんばってバージョンアップを果たしてもらいたい。（T）

レイトレーシングにおいて半影を生成する

HASH.X

GRAPHICS

Ichien Toru 一圓 亨

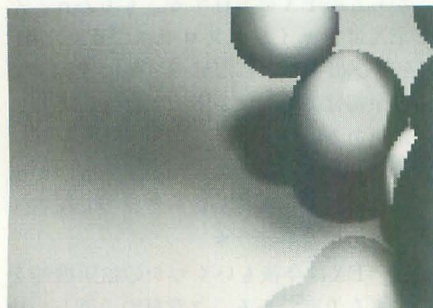
1月号のディスクに収録されていた半影つきレイトレーシングシステムHASHです。コンパイル方法やアルゴリズムなど，1月号で十分に説明できなかった分を今回まとめて解説します。ソースリストなどは1月号のディスクを参照してください。

レイトレーシングの立場

レイトレーシング……「力」で3次元空間の数式を目に見える形に変換するアルゴリズム。しかし残念ながら，レイトレーシングがCMや映画の1シーンなどに使われることはあまりないようだ。

その最大の理由は，レイトレーシングが宿命的に持つ計算時間の問題だろう。計算時間だけで選ぶなら，Zバッファ法*1やスキャンライン法などのほうが明らかに速い。まして，Phongシェイディングなどのシェーディング補間法（スムースシェイディング）*2でも使われた日には，（反射や屈折は別として）レイトレーシングと変わらない画像を，レイトレーシングの何倍も短い時間で生成できるのだ! CMや映画で使うとなるとほんの数秒のアニメーションで何十枚もの静止画が必要となるわけだから，1枚の描画に何時間もかかるレイトレーシングではアニメーションに向かないのだろう。

では，「レイトレーシング」の立場はどうなるのだろう。実用的でないアルゴリズムとして，消え去るしかないのだろうか？……この答えは簡単に出すわけにはいかない。実際に，レイトレーシングの強力な表現力のゆえに，膨大な時間をかけて作成されたアニメーションもあるのだ。

前置きが長くなったが，ここで今回のテーマである。レイトレーシングがその膨大な計算時間と引き替えに手に入れた，強力な「表現力」をさらに強化してしまおうというのだ。しかも，計算時間の悪化を最小限に抑えて。

滑らかな影

＊1　Zバッファ法については参考文献2)を参照してほしい。
＊2　Phongシェーディングについても参考文献2)を参照してほしい（手抜きか？　いやいや，私の専門外なので……）。

なにを強化するのか？

これまでにレイトレーシングの作品（画像）を見て，なにか不自然に思ったことはないだろうか？「CGなんて全部不自然だ」などと意地悪をいわないように(ヨロシク)。

不自然なのだ。影が。なんであんなにシャープなエッジができるんだ？　現実世界の影はエッジがぼんやりしていて，影の境界線を定規で引くようなわけにはいかないはずだ。レイトレーシング（に限らず，多くのCGのアルゴリズム)で，シャープなエッジを持った影が見られるのは，光源が大きさを持たない「点光源」として扱われているためである。

よし，影処理を強化しよう。でもどうやって？　まず頭に浮かんだのは分散レイトレーシング*3だ。分散レイトレーシングなら，光源が大きさを持つために生ずるぼやけた影(半影)を表現できる。しかし，分散レイトレーシングでは「計算時間の悪化を最小限に」という条件にあてはまらない。このとき，ふと頭に浮かんだアルゴリズムは，かなりの近似と扱える形状の制限を受けつつも十分現実的な半影を生成できると推察された。さっそくプログラミングにかかり，'HASH'をものの2週間で完成してしまった。ちなみに，'HASH'という名前は，"HAlf-SHadow ray tracing system"からつけたものだ。

＊3　分散レイトレーシング(distributed ray tracing)，分配レイトレーシングと呼ばれることもある。アイデアは，ひとつのピクセルを計算するのに，そのピクセル内にK本の視線(レイ)をランダムに発生させ，ピクセルの色をK本のレイの色の平均として求めようというものだ。分散レイトレーシングはアンチエイリアシング効果を持っているのだ。また，レイの分散（分配）は1次レイだけでなく，2次以降のレイに対しても適応される。すなわち，なんらかの物体に当たって反射するレイもまた，正反射方向を中心にしてランダムにL本分散されるのである。これによって，金属表面の鈍い映り込みを表現することができる。

これと同じことが屈折光にもいえて，この場合には曇り硝子のような鈍い透過を表現できる。肝心な影処理だが，これもやはり，大きさを持った光源に向かってM本のシャドウフィーラをランダムに発生させて，M回の影処理を平均することによって半影を生成する。ほかに，時間軸に関してランダムにN点サンプリングし，N点の平均を取ることによって動きのブレを捕らえること（モーションブラー）も分散レイトレーシングでは可能である。分散レイトレーシングは，アンチエイリアシング・鈍い反射・鈍い透過・半影・モーションブラーと，正に究極のレイトレーシングである。しかしながら，ランダムに発生させたレイの平均でピクセルの色を求める以上，K，L，M，Nの値はある程度大きくなければならない。したがって，レイの数は極端に増加し，計算時間も極端に長くなるのである。実際には，工夫を凝らしてより少ないレイで実現しているらしい。

半影を生成するレイトレーシング

レイトレーシングの中で半影を生じるアルゴリズムには，前節で紹介した分散レイトレーシングのほかに円錐（コーン）レイトレーシングがある。これは1984年にJ．Amanatidesによって発表されたアルゴリズムで，レイ（視線：直線）の代わりに円錐ビームを使う方法である。この方法も分散レイトレーシングと同様に，アンチエイリアシング・鈍い反射・鈍い透過・半影などの効果を持つ。

分散レイトレーシングにせよ，円錐レイトレーシングにせよ，演算時間が通常のレイトレーシングの数倍から十数倍になることは必至で，10MHzの68000で実現するのは難しい。今回のプログラムでは，アンチエイリアシングも鈍い反射も鈍い透過もモーションブラーも表現できないが，半影だけは高速に表現できる。

一般の影処理(全影処理)のアルゴリズム

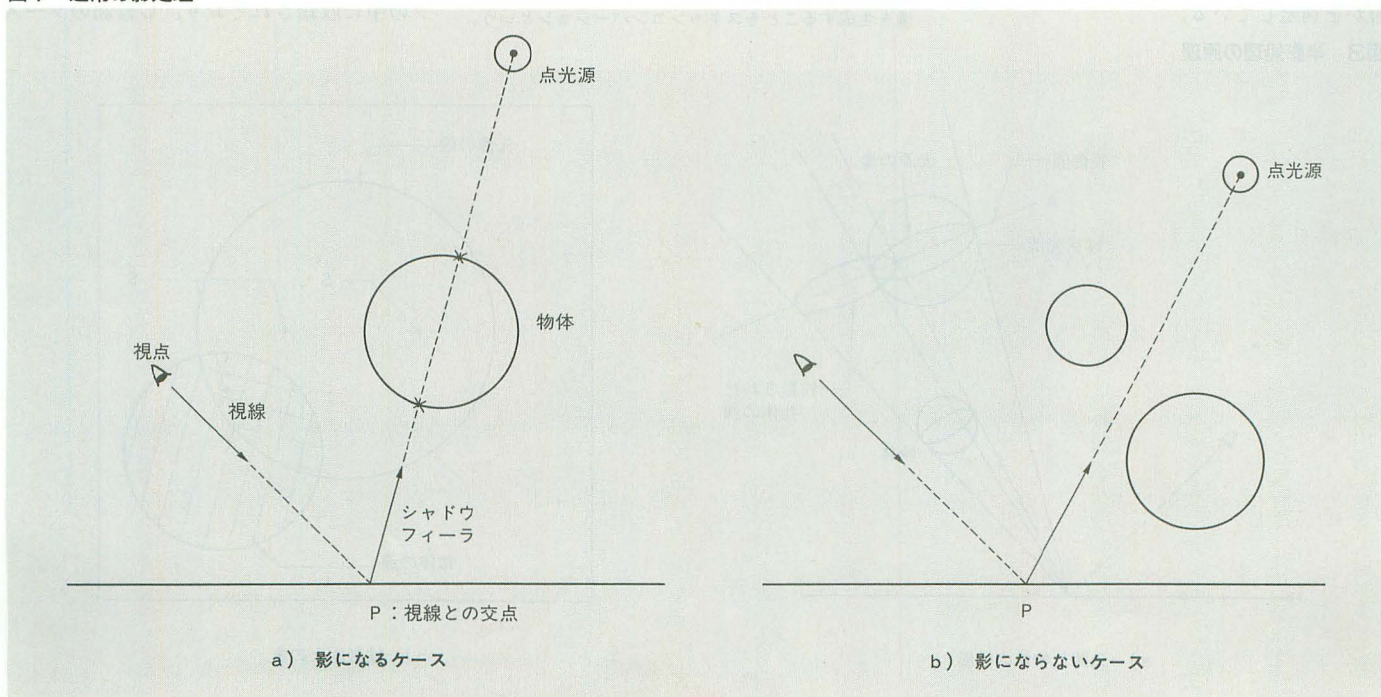
すでに述べたとおり，一般のレイトレーシングでは光源として，大きさを持たない点光源を想定している。したがって，影処理は次のように行われる。

ある物体と視線との交点がもとまると，この点から点光源へ向けてシャドウフィーラと呼ばれる視線を発生し，シャドウフィーラとほかの物体との交点を調べる(図1)。もし交点があれば，光源からの光が交差した物体によって遮られることになるから，もとの点は影になる。

この方法は，シャドウフィーラと物体との交点が「ある」か「ない」かによって影を生成するので，一様な影となり半影は生じない。

半影処理のアルゴリズム

現実世界で半影を生じる原因について，光波の回折によるものなどいくつか挙げることができる。今回の半影処理で対象となっているのは光源が大きさを持つために生ずるもので，光源として球状の物体を想定している(図2)。

いま，ある物体と視線との交点：Pを考え，P点が影になるかどうか調べるためP点と球状光源の中心を結ぶ線に垂直に投影面：Aを設定する。P点を視点，A面を視野面とする隠面問題を解決できれば，A面に投影された光源の面積：Sと他の物体に隠されて残った部分の面積：S'を知ることができる(図3)。光源の明るさをI_{LIT}とするなら，P点における光源光の強度：I_{LIT-P}は次式により得られる。

$$I_{LIT-P}=I_{LIT}\times(S'/S)$$

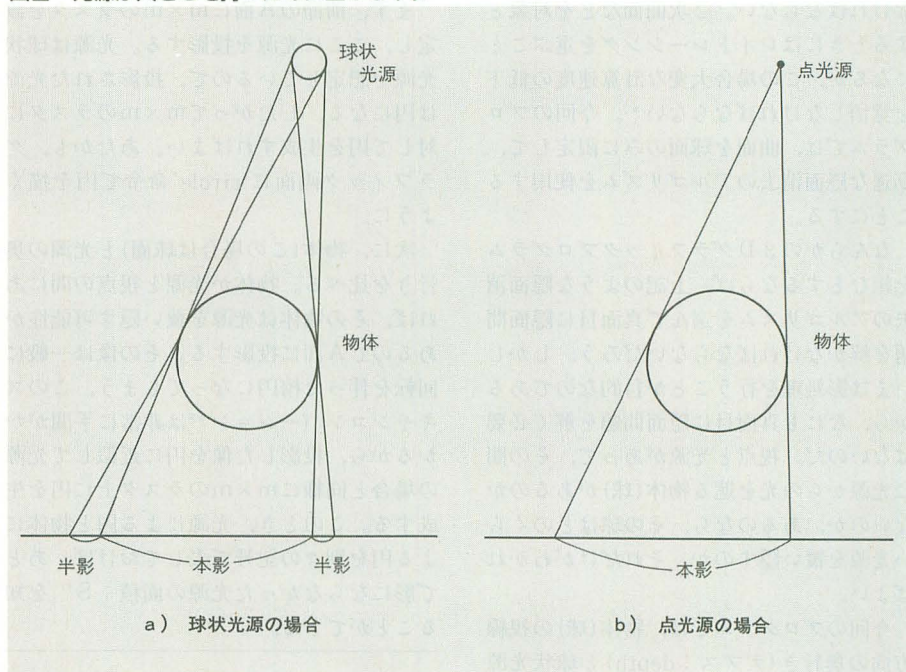
プログラムでは球状光源の各部からくる光はいたるところ一定と仮定している。実際には光源の中心付近からくる光と周辺部からくる光とでは強度が違うだろうから，投影面上の面積：S'について積分しなければならない。さらに，光源の中心付近と周辺部からの光とではP点への入射角が異なるため，正確なレンダリング*4を行おうとすると「陰」処理(shading)に対してもS'について積分しなければならない。しかし，これらはいたずらにレンダリングの計算速度を低下させるうえに，たいした効果も期待できないので今回は目をつぶった。

*4　簡単に説明すると，つやなど物体の質感を含めた色を決定する過程のことをいう。

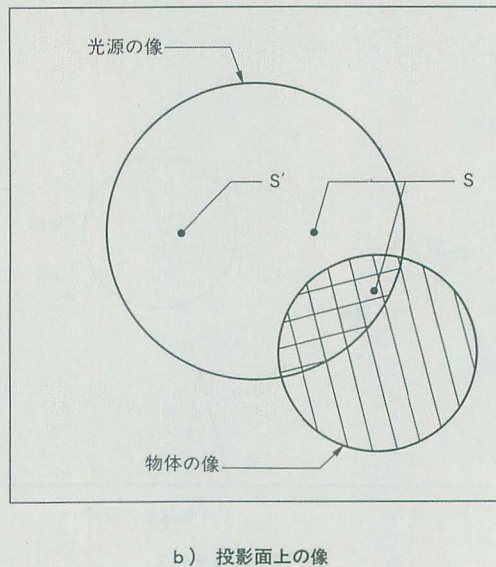
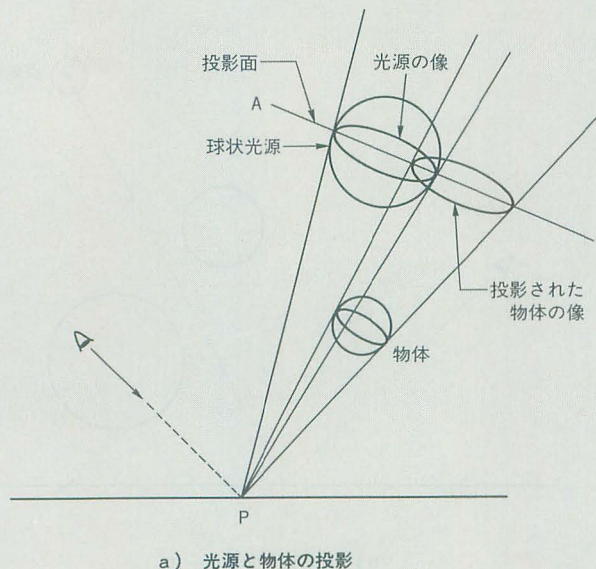
図2　光源が大きさを持つために生じる半影

図1　通常の影処理

隠面消去問題の解法

昔から隠面問題*5を解決するアルゴリズムについてさまざまな方法が考えだされてきた。もちろんレイトレーシングもその方法のなかのひとつである。ほかにZバッファ法・スキャンライン法・リストプライオリティ法・奥行き法など，さまざまな方法がある。今回も半影処理において投影面：A上にとったm×mのラスタ*6について隠面消去を行うためにいずれかの方法を選ばなければならない。2次曲面などを対象とするときにはレイトレーシングを選ぶことになるが，この場合大変な計算速度の低下を覚悟しなければならない*7。今回のプログラムでは，曲面を球面のみに限定して，高速な隠面消去のアルゴリズムを使用することにする。

なんらかの3Dグラフィックプログラムを組むとするならば，上記のような隠面消去のアルゴリズムを選んで真面目に隠面問題を解かなければならないだろう。しかしいまは影処理を行うことが目的なのであるから，なにも真面目に隠面問題を解く必要はないのだ。視点と光源があって，その間に光源からの光を遮る物体(球)があるのかないのか，あるのなら，その球はどのくらい光源を覆い隠すのか。それだけがわかればよい。

今回のプログラムでは，物体(球)の視線方向の奥行き(デプス：depth)と球状光源のデプスとを比べてその物体が影を作るか否かを判定している。

*5 正確には「隠面消去問題」。ある点から物体を見たときにどの面が見えるか，あるいは見えないかを判定し，見える面だけを表示すること。
*6 点描画面のことをラスタという。
*7 この場合，力づくでm×mのラスタについてレイトレーシングを行うのもよいが，それではとても実用にならないので，実際には分散レイトレーシングに頼ることになるだろう。

光源の投影ー球面の投影＝半影

ここで，今回の半影処理の手順を具体的に示そう。

まず，前節のA面にm×mのラスタを設定し，ここに光源を投影する。光源は球状光源を想定しているので，投影された光源は円になる。したがってm×mのラスタに対して円を生成すればよい。あたかも，グラフィック画面に'circle'命令で円を描くように。

次に，物体(この場合は球面)と光源の奥行きを比べる。物体が光源と視点の間にあれば，その物体は光源を覆い隠す可能性があるのでA面に投影する。その像は一般に回転を伴った楕円になってしまう，このスキャンコンバージョン*8は非常に手間がかかるから，投影した像を円に近似して光源の場合と同様にm×mのラスタ上に円を生成する。このとき，光源による円と物体による円を別々の記号で表しておけば，あとで影にならなかった光源の面積：S' を知ることができる。

*8 直線や円などを画面上のドットに近似する手順をラスタ化といい，スキャンラインの順序で画像を生成することをスキャンコンバージョンという。

なお，円のスキャンコンバージョンには「Bresenhamの増分法」を用いた。

光学モデル(拡散反射)

拡散反射には，Lambertの拡散反射モデルを使用した。

$I_{DIF}=C_{PRM}\times I_{LIT}\times(L\cdot N)$

I_{DIF}：拡散反射強度
I_{LIT}：光源光強度
C_{PRM}：物体の色
L：　光源の存在する方向
N：　法線ベクトル
ただし，$|L|=|N|=1$
(L・N) は，LとNの内積，
|L|はベクトル：Lの大きさを表す。

鏡面反射(ハイライト)

鏡面反射には，Blinnの鏡面反射モデルを使用した。

$I_{SPC}=R\times I_{LIT}\times(H\cdot N)^{G}$

I_{SPC}：鏡面反射強度
R：　反射率
G：　光沢(大←つやあり←20→つやけし→小)
H：　$H=(V+L)/|V+L|$
V：　視点の存在する方向
ただし$|H|=|V|=|L|=1$

半影レイトレーシングの実験

プログラムはすべて1月号の付録ディスクの中に収録されており，C言語のソース

図3　半影処理の原理

a） 光源と物体の投影

b） 投影面上の像

プログラムは5つのモジュールより成り立っている。

1) メインモジュール　　　　:hash.c
2) サブモジュール　　　:hash_sub.c
3) ディファレンスアブソーバ:differ.c
4) フレームバッファドライバ:frame.c
5) ベクトル演算モジュール　:vector.c

プログラム全体の規模からみると，かなり大きいように思えるが，実はレイトレーシング本体は'hash.c'と'hash_sub.c'の2つのモジュールだけで，残る3つのモジュールは，

●differ.c
　システム，ライブラリなどの「違い」を吸収する
●frame.c
　グラフィックの取り扱いを汎用化する
●vector.c
　3次のベクトル演算を行う

のように，汎用になっている。半影を生成するレイトレーシングとしてはかなりコンパクトにまとまっているのではないだろうか。

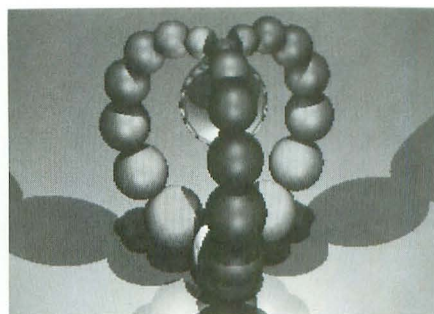
なお，'frame.c'，'vector.c'，'differ.c'に関しては，苦労して汎用に組んだものなので愛機X68000のためにも，多くのユーザーに使用してもらいたいと思う。ついては，Oh!Xへの投稿に関しては「一圓亨の'vector.c'を使った」とひと言書いてもらえれば使用は自由である。もちろん，個人的な使用に関してはこの限りではない。どんどん使ってほしい。

コンパイル

コンパイルするには，5つのソースファイルに各モジュールの定義ファイル（hash.h, frame.h, differ.h, vector.h）を加えた9つのファイルが必要である。ここでは上記のファイルが適当なディレクトリにあって，カレントパスがそのディレクトリまで通っているものとする。コンパイラはGCCまたはOS-9上のMW-Cを使用する。

GNU-C:
```
gcc hash.c hash_sub.c frame.c differ.c vector.c -lbas
```
OS-9/X68000+MW-C:
```
cc hash.c hash_sub.c frame.c differ.c vector.c
 -l=lib/xos9_pss.l
```

各コンパイラとも，最小限必要なオプションのみ示した。各自必要に応じてほかのオプションを付けてほしい。なお,「XC ver 1.0」のバグのため「XC ver 1.0」ではコンパイルできない。「ver 2.0」については手元にないので未確認である。もし，XCしか手元になくて，それでも'HASH'を使いたいという，嬉しい読者がいたら，誠に申しわけないが「人間プリプロセッサ」をしてもらいたい。ここで問題になっているXCのバグというのは条件付きコンパイル命令の周辺のものなので，識別子"__XC__"が定義されているときにコンパイルされるべき部分を残し，それ以外の余計な#if〜#endifをエディタなどで削除してコンパイルすればよいはずだ。

ただ，この方法も確認したわけではないので，実際に「人間プリプロセッサ」を実行しても素直にコンパイルできるとは限らない。その辺は，自分の責任においてやってほしい。本音をいえば，私もあのXCでデバッグをするなどという精神衛生上よろしくないことはしたくないのだ。XCをすでに持っている読者なら，Oh!X(1990年6月号)でGCCを手に入れたはずだし，この号を買い忘れてしまった人もなんとかGCCを手に入れたほうが無難だと思う。

ハート曲線の組み合わせ

実行方法

'hash'に続いてデータのファイル名を打ち込み，リターンキーを押すとレイトレーシングを開始する。

　hash 'データのファイル名'

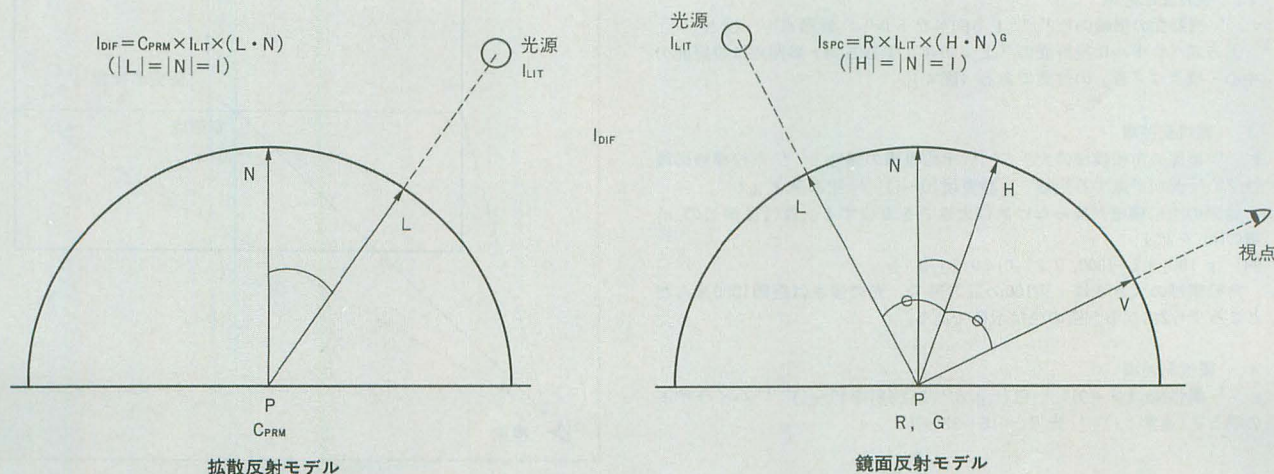
もし，完成した画像をセーブしたいならデータのファイル名に続いてセーブするときのファイル名を記述する。

　hash 'データのファイル名' 'セーブ時のファイル名'

セーブは，Human68kのベタフォーマットで行われるのでX-BASICなどから簡単に読み込める。また，'hash'で画面をロードすることも可能である。

　hash frame load 'ロードするファイル名'

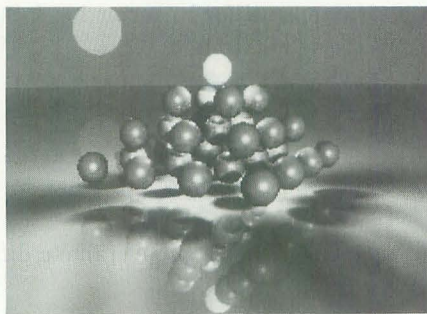
写真を見ればわかるとおり，通常の影処理を行った場合の影は，非常にシャープなエッジを持っていて，影となるところは至るところ一定の暗さとなっているのがわかるだろう。画面左上に見えている丸い物体

図4 鏡面反射モデル

従来のレイトレ

HASHではこうなる

は無限遠方にある光源で，この光源による薄い影も確認できる。これに対し，半影処理を施した画像では，完全に暗い部分(本影)とその周りにぼやけた部分(半影)が見られる。現実世界の光源は必ず大きさを持っており，写真に示すようなシャープな影ができることはない。

今後の発展

完成画像を見た感想はいかがだっただろうか。扱える曲面が球面のみに限られているとはいえ，こと影に関してはかなり現実感のあるものが得られているのではないだろうか。今後，このプログラムを発展させようとするとき，まず考えられるのは曲面の種類を増やすことだろう。

レイトレーシングでは2次曲面が扱えるのが普通だから，半影処理も2次曲面に対応させなければならない。しかし，対象が2次曲面となると，今回用いた簡単で速いアルゴリズムは使えなくなる。それでも，2次曲面に対応させたいのなら，あとは分散レイトレーシングか円錐レイトレーシングに頼ることになる。

ほかに，非常に特殊ではあるが，多面体(3角パッチ)に対してレイトレーシングを行うアルゴリズム*9がある。影処理のときにZバッファ法などを使って，多面体を構成する3角パッチを投影面に投影してやれば今回のような半影処理を実現できる。レイトレーシングにおいて今回の半影処理を発展させるとすればこの方向しかないだろう。

多面体であればZバッファ法などのほうが高速だし，スムースシェーディングが使えて有利なのだが，多面体のレイトレーシングというのもなかなか面白そうなテーマなので，そのうち機会があれば挑戦したいと思っている。

*9　興味のある方は文献7.4）　をどうぞ。

おわりに

半影処理された画像を見て，我ながら感動してしまった。自作の別のレイトレーサ

データの文法

レイトレーシングのデータの中には，次のものが含まれる。

1．　システム記述項

s　‘横ピクセル数’‘縦ピクセル数’‘ビット数(4, 8, 16)’‘ディザリングモード(0～4)’‘左上X座標’‘左上Y座標’‘右下X座標’‘右下Y座標’‘最大追跡回数’‘影処理用ラスタのメッシュ’

ピクセル数とビット数によって画面モードが決定される。

最大追跡回数は反射光の追跡を行う最大値を与える。

メッシュ：meshは半影処理用ラスタの細かさである。大きな値とすると緻密な半影を生成するが若干計算に時間がかかる。mesh＝1とすると通常の影処理を行い，mesh＝0ではまったく影処理を行わないため，計算が早く終わる。

例）s 512 512 16 0　0 0 511 511　3 30

mesh＝30の半影処理を行いながら，512×512(65536)モードのフルスクリーンのレイトレーシングをする。

2．　視野面記述項

v　‘視野面の横幅の1/2’‘上方向ベクトル’‘参照点’‘視点’

上方向ベクトルは視野面の「上」が存在する方向・参照点は視野面の中心・視点は「目」の位置である（図4）。

3．　環境記述項

g　‘地面の市松模様の大きさ’‘市松模様の属性1’‘市松模様の属性2’‘光が半減する距離’‘霞表現(0～1)’‘空の色(r, g, b)’

地面の市松模様が要らなければ大きさを0にする。属性は後述の a' 項のNo.を記す。

例）g 100. 1 2　1000. 0.2　0.4 0.6 1.0

市松模様の大きさは1辺100の正方形で，光の強さは距離1000進んだところで1/2になる。空の色は水色である。

4．　属性記述項

a　‘属性No.(＞＝0)’‘色(r, g, b)’‘反射率(0～1)’‘ハイライトの明るさ(通常：1)’‘光沢(～16～32～)’

属性No.は球面や地面の市松模様の属性を決定するのに用いられるNo.である。

5．　光源記述項

l　‘光体の属性(＞＝0)’‘中心座標’‘半径’‘光源光強度’

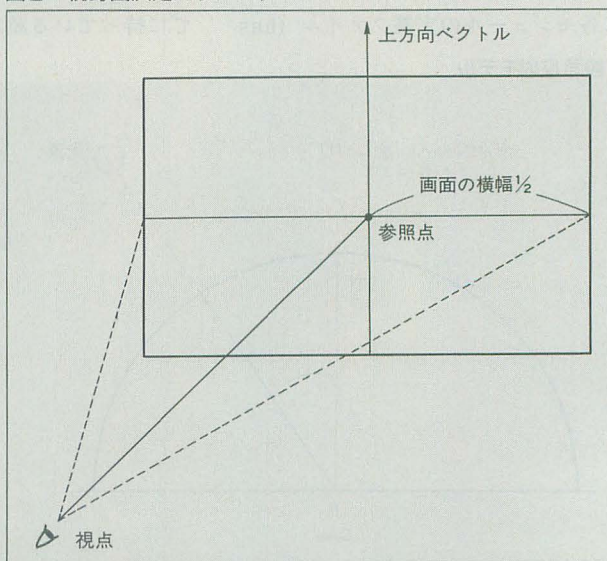
l　‘－1’‘光源の存在する方向’‘見掛けの大きさ（単位：度)’‘光源光強度’

属性に対して負の数を与えると無限遠方からの平行光線源の定義と見なされる。そうでない場合は有限な位置にある球状の光源と見なされ，指定された属性を持つ実体が同時に定義される。

半径および見掛けの大きさは0であってはならない。

6．　球面記述項

p　‘球面の属性’‘中心座標’‘半径’

図5　視野面決定パラメータ

では，2次曲面や屈折などもサポートしているのだが，影がノッペリしているため，「ううむ……」と唸ってしまう。おかげで分散レイトレーシングをやりたくなった。しかし，X68000では遅すぎる。ここは我慢だ。学校でコプロ付きのP○-98RAのスピードなんぞを見せつけられると，つい我が家のプリンタの下に敷いて計算端末にしたい欲望に駆られてしまう。X68000は，このスピードで厳しい生き残り競争を勝ち抜いていけるだろうか。

それはそうと，X68000のプログラミング環境は最高である。グラフィック・スプライト・PCM……，X68000にはプログラム欲を駆り立てるテーマが山ほどある。まして，OS-9がある。今回のプログラムはすべてOS-9/X68000上のMICROWARE-C(MW-C)で組んだものだ。MW-CとXCのライブラリの違いのため，これまで(hash以前)のプログラムは移植を渋っていたのだが，近頃，GNU-C・FLOAT3＋・コプロボードを手に入れたことからGNU-Cへの移植に踏み切った。

1年半前，私がOh!FMではなく，Oh!Xを買うようになっていちばん寂しかったのは，OS-9の記事がほとんどなかったことだった。いまではOS-9は開発の環境なのだと割り切っているので，もう「寂しい」とは思わなくなった。私にとっては，裏でレイトレーシングをしながら表ではエディタでデバッグができる環境(OS-9)はとても心地好いのだ。

参考文献

1)　小林一也，「レイトレーシングを用いた球面の3次元グラフィックス」，ソフトバンク，Oh! FM　1984年5，6月号

2)　丹明彦，「Zバッファアルゴリズム」，ソフトバンク，Oh!X，1989年7，8月号

3)　デビッド．F．ロジャース，「実践コンピュータグラフィックス　基礎手続きと応用」，監修：山口富士夫，訳：セイコー電子工業[株]電子機器事業部，日刊工業新聞社

4)　高桑昌男，「CGレイトレ物語」，アスキー出版局

5)　中前栄八郎，「ここまできたリアリズム－四半世紀の進展－」，図形処理情報センター，PIXEL，1989年2月号

リスト

```
==================  heart  ==================
 1: *  H e a r t - s h a p e d   d e c o r a t i o n  *
 2:
 3: *  wid hgt bit dth    x1 y1   x2  y2     dx dy  max  mesh  *
 4: s  512 512  8   4     0  0  511 511      4  6   3   40
 5:
 6: *  画面幅/2  上方向     参照点          視点  *
 7: v    320      0 0 1    0 0 200        860 -80 700
 8:
 9: *  No   red  grn  blu  ref high blinn  *
10: a   0  1.00 1.00 .200  .35   3   36
11: a   1  .500 .500 .500  1.0   0   48
12:
13: *  chek at1 at2  half dift amb      sky color  *
14: g  0.   0   0     0.   0.  .00      .40 .20 .30
15:
16: *  atr    x  y    z     r      light intensity  *
17: l   1     0. 0. 290.   60.       .50 .50 .50
18:
19: *  par    x  y    z     r     light intensity  *
20: l  -1     1. 0.   0.    1.     .60 .60 .60
21: l  -1     1. 0.   3.    4.     .60 .60 .60
22:
23: *  No   red  blu  grn  ref high blinn    atr    x y   z        r  *
24: a   2  1.00 1.00 1.00  .01  30    32  p  2     0 0 350     17.7471
25: a   3  1.00 1.00 1.00  .01  30    32  p  3     0 0  50     50.2430
26:
27: *  No   red  blu  grn  ref high blinn    atr     x      y    z        r  *
28: a   4  1.00 .000 .000  .01  30    32  p  4  83.2336 0 89.0560  45.9707
29: a   5  .830 .170 .000  .01  30    32  p  5  137.886 0 152.114  41.7228
30: a   6  .670 .330 .000  .01  30    32  p  6  159.369 0 223.987  37.5075
31: a   7  .500 .500 .000  .01  30    32  p  7  150.000 0 290.000  33.3372
32: a   8  .330 .670 .000  .01  30    32  p  8  117.795 0 338.792  29.2321
33: a   9  .170 .830 .000  .01  30    32  p  9  74.2462 0 364.246  25.2209
34: a  10  .000 1.00 .000  .01  30    32  p 10  31.5714 0 366.220  21.3602
35:
36: *  No  red  blu  grn  ref high blinn   atr      x         y        z        r  *
37: a  11 .000 1.00 .000  .01  30    32  p  11 -41.6168 -72.0824 89.0560  45.9707
38: a  12 .000 .830 .170  .01  30    32  p  12 -68.9430 -119.413 152.114  41.7228
39: a  13 .000 .670 .330  .01  30    32  p  13 -79.6845 -138.018 223.987  37.5075
40: a  14 .000 .500 .500  .01  30    32  p  14 -75.0000 -129.904 290.000  33.3372
41: a  15 .000 .330 .670  .01  30    32  p  15 -58.8975 -102.013 338.792  29.2312
42: a  16 .000 .170 .830  .01  30    32  p  16 -37.1231 -64.2991 364.246  25.2209
43: a  17 .000 .000 1.00  .01  30    32  p  17 -15.7857 -27.3416 366.220  21.3602
44:
45: *  No  red  blu  grn  ref high blinn    atr      x        y        z        r  *
46: a  18 .000 .000 1.00  .01  30    32  p  18   -41.6168 72.0824 89.0560  45.9707
47: a  19 .170 .000 .830  .01  30    32  p  19   -68.9430 119.413 152.114  41.7228
48: a  20 .330 .000 .670  .01  30    32  p  20   -79.6845 138.018 223.987  37.5075
49: a  21 .500 .000 .500  .01  30    32  p  21   -75.0000 129.904 290.000  33.3372
50: a  22 .670 .000 .330  .01  30    32  p  22   -58.8975 102.013 338.792  29.2312
51: a  23 .830 .000 .170  .01  30    32  p  23   -37.1231 64.2991 364.246  25.2209
52: a  24 1.00 .000 .000  .01  30    32  p  24   -15.7857 27.3416 366.220  21.3602
53:
54: e
```

製品試用レポート

Fine Scanner-X68

Takahashi Tetsushi 高橋 哲史

HAL研究所の「Fine Scanner-X68」はモノクロ256階調の取り込みが可能なユーザー待望のハンディスキャナです。添付ソフトも強力でグラフィック作成のおともに最高と高橋君も絶賛。さっそく使用感をレポートしてもらいましょう。

こんにちは。お絵描き専門スタッフの高橋くんです（笑）。さてさて，今日は全国1千万人（当社推定？）のCG描きの皆さんに朗報です。そう，CGの下絵を用意してスキャナで取り込んでは，うまくいかず，何度もコピー機とスキャナの間を往復したあげく「だあーっもうやってらんねーっ！」（CG描きなら誰でも一度は経験があると思う……）となってしまったあなた！HAL研究所から出たんですよ，賢いスキャナが！　これで，いままでの取り込み調整の負担がかなり軽減されることと思います。とにかく賢く，使って嬉しくなってしまう私なのでした(ほくほく)。ああ，こんなに可愛いやつなのに，この原稿書き終わったらまた編集部に返さなくてはならないなんて。うう，なんて非情な世の中なの(編注：当たり前です。欲しかったら自分で買いなさい。それが人生です)。

製品概要

さて，このファインスキャナがいままでのハンディスキャナとどこがどう違う（賢い）かといいますと，なんといっても「グレイスケール256階調取り込みを実現している」という点なのです。これにより元絵のペンの強弱（いわゆる「入り」と「抜き」）までがきっちりとディスプレイ上で再現してくれます。はっきりいってこれは感動ものです。いままでのハンディスキャナでは白と黒2階調のみの取り込みでしたので，線がプチプチと切れてしまい，その修整にたいへん時間を要してしまいましたものね（元絵を描いている時間より修整してる時間のほうが長かったりして）。

それと専用の画像加工ツール「Image Photo 68」で取り込んだ絵をディスプレイ上でリアルタイムにいじれますし，まさにいたれりつくせりといった感じがあります。いままでのスキャナではせいぜいBASICの取り込みプログラムがサンプルでついてくる程度でしたからこれはもう天と地ほどの差です。

さて使い心地は？

読み取り幅や転送速度などの詳しいスペックは1月号の広告を見れば載っていますので，ここでは付属ソフトである「Image Photo 68」を詳しく見ていきたいと思います。このソフトでは明るさやコントラストなどの細かい調節がマウスひとつで簡単に行えるようになっています（もちろんキーボードからの操作も可能）。また複写や回転などの簡易グラフィックツールとしての機能も兼ね備えているのでかなり重宝することでしょう。

さて具体的に取り込みはどう行うのでしょうか？　このソフトは「1024×1024」の取り込み画面を持ち，その中から範囲を指定してそこに画像を取り込むようになっています（もちろん最大取り込み面積は「1024×1024」です！）。あ，いい忘れていましたが画面モード設定により768/512ドットの両モードに対応しています。そして取り込みには「範囲内取り込み，自由長取り込み」の2種類があり，範囲内取り込みには，

256×256　512×512
768×512　640×400
512×480　全領域（1024×1024）

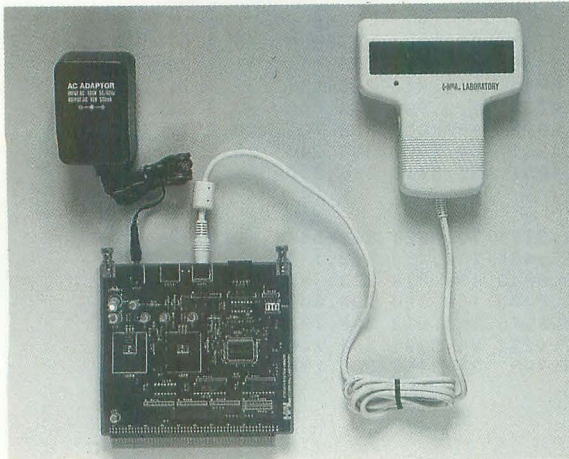
での指定が1024×1024の画面の任意の点から行えるようになっています(便利ですねー)。これにより複数の絵を同時に取り込み，画面上で編集することが可能になっています。また自由長取り込みは画面モードに関わらず1024×1024の取り込みを行います。

特筆すべきは取り込みの速度で，本当に「スルスルー」と画面が滑るように流れ，あっという間に取り込みが完了します（あまり調子にのって速く転がしすぎると下絵を読み飛ばしてしまうのでご注意を)。これなら元絵の位置がずれていたといっても，取り込み直しが苦になりませんね。う～ん，専用ボード付きで本当によかった。

さて，取り込んだ絵はその場ですぐに加工ができます。コントラスト調整や2値化など，さまざまな機能がありますが，とりあえず順番に見ていくことにしましょう。

まずはメニューバー3番目の「調節」です。ここではコントラスト，明るさの調節ができますが，なんといっても重宝するのはγ補正です。これは出力先のデバイス(プリンタやディスプレイですね）に合わせて輝度を自動調節してくれるという非常に便利な機能なのです。プリンタもドットインパクト，熱転写，レーザーと多種多様に対応しており思わず感心してしまいました。ちなみにデフォルトは「HALFAX」になっていました（笑)。

そしてさらに強力なのが「ユーザー曲線によるγ補正」です。マウスでちょいちょいとユーザー曲線をいじるだけで画面の色合いが瞬時に変わっていくのは実に爽快です（768モードでは「設定」をクリックしないと変化しないんですけどね）。これは次の2値化の細かい調整に威力を発揮します。CG描きには非常にありがたい機能なのです（こだわりで自分の求める線が得られるのが良！）。

そしていよいよ肝心要の2値化です。2値

化は文字どおり表示されているグラフィックを白と黒の2色だけに置き換えていく作業なのです。「Image-Photo-68」ではこの2値化にかなり力を入れており単純2値化を始め誤差拡散，ベイヤーなどさまざまな手法が用意されているので，取り込んだ画像にあった2値化を施すことができます(ちなみに自分で作成したディザパターンも使えます)。でも僕が主に使うのは単純2値化だけなんですけどね。いわゆるアニメ調のCGばっかり描いているもので。しかし写真などの取り込みにも威力を発揮するので，DTP（デスクトップパブリッシング）にも非常に重宝するのではないでしょうか（もちろんソフトがあればの話ですが……。X68000のDTP環境はとても不幸だと思う私)。

また1024×1024の任意の範囲を倍率を変えて2値化を施すことができるので，たとえば1024×1024全域に取り込んだ絵を512×512に収まるようにすることもできるのです(縮小率50%)。ただ512×512だとディスプレイ上でのドットの縦横比が1：1ではないので，1024×1024の絵を正直に縮小率50%で持ってくるとやや潰れた絵になってしまいますので注意したいところです（お勧め倍率は横50%縦37%といったところでしょうか？)。まあ，この辺は画面表示でリアルタイムに2値化を施した画像を確認できるのであまり心配しなくても大丈夫でしょう。

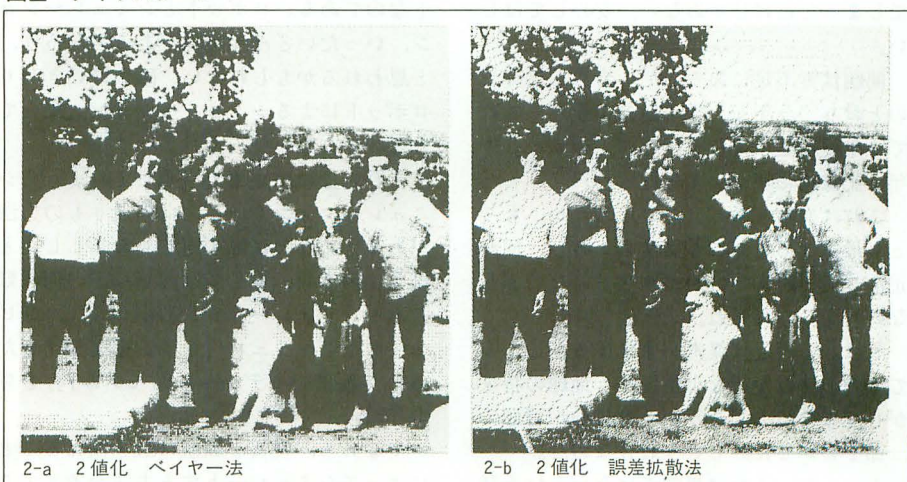
さて2値化した画像を保存します。2値化された画像に限らず「Image Photo 68」は多彩な画像フォーマットをサポートしています。驚くことに市販ツールとして初めてPIC形式のファイルをサポートしています（おお！)。ただ私が使ったものはサンプル版のせいかまだPICのPの字も見えませんでしたが……（しかし広告には明記してあるので製品版にはきっと入っているでしょう)。やはり，いまとなってはPICがX68000の画像フォーマットの標準となった感がある昨今，ほかのツールの皆さんにも見習ってほしいものです。

あと，ちょっと難をいいますと，ディスクへの書き込み&読み込みが少々遅いです。GL3など平気で1分以上かかったりします。ほかがいいだけに，ついついいってしまう私です。

さて，そのほかの点はきちんとツボを押さえて作られており，プリンタにも幅広く対応していますし，設定すればなんとビデオスキャナも使えます。少々残念な点としては512×512モード多値のファイル書き込みでは画面で見たどおり少し縦に潰れたままSAVEしてしまいます。512×512が一番よく使うモードでありますし，この辺はもう少し頑張ってほしかったのですが……。

それと，基本的にマルチウィンドウなのですがそれぞれのウィンドウの位置が固定されているのでやや不自由さを感じます。特にユーザー曲線の定義などは実際に表示されている絵を見ながら行いたい作業ですので，設定中でもウィンドウやスクロールバーが動くようにしてもらいたかったものです。しかし，Z'sSTAFF PRO-68Kやマジックパレットなどのグラフィックツールのスキャナウィンドウよりはずっとかゆいところに手が届く出来になってるのが嬉しいところです。

これは使える

ハンディスキャナというと，これまでは結構ちゃちな物というイメージがあったのですが，このファインスキャナは見事にそのイメージを払拭するだけの性能を持っています。1月号付録ディスクのカット取り込みのときに初めて触ったあの感動はちょっと言葉にしがたいものがあります。とにかく取り込み画像が綺麗なのです。もちろん，CZ-8NS 1などの机置き型スキャナ（そんな名称あったっけ!?）に比べると「カラー取り込みができない」「あまり大きなサイズの絵は読み込めない」などの点があることはありますが，カラー取り込みなどを使う人などは滅多にいないし（レイトレのマッピングデータを作るくらい？），ハンディタイプのメリットである小回りのよさがそれを補って余りあると私は思うのです。それになんといっても39,800円という価格設定が魅力的です。高速取り込みのための専用ボード，賢い付属ソフトつきでこの値段は絶対安い！

ということでこのファインスキャナはお勧めですので全国2千万人の(おい増えてないか？)CG職人の皆様のお手元にぜひこの1台を（なんか通信販売の宣伝みたいになってしまいましたね)。このスキャナの登場により良質のCGが限りなく増えることを願う私でした。

Fine Scanner-X68　HGS-68　39,800円
（専用ボード，ソフト付き）
HAL研究所　☎03(3252)5561

図１　ペン画の取り込み例（単純2値化）

1-a　原画　　1-b　プリントアウト

図2　ディザパターン

2-a　２値化　ベイヤー法

2-b　２値化　誤差拡散法

シミュレーションは未来をひらく

今回は「連続変化型シミュレーションに挑戦」ということで，電熱式温水器の制御を取り上げてみました。と，本題のほうはそういう内容なのですが，前半部分はシミュレーション関係の書籍の紹介となっていますので，興味のある方はどうぞ。

Kamon Masato 華門 真人

正確にいえば，ではあるが，'90年代というのは1991年から始まる10年間のことである。つまり，ようやく名実ともに'90年代を迎えたということになる。

ここ数年で世界は大きな変化を遂げた。統一ドイツが誕生し，アメリカがソ連に最恵国待遇を与えるようになるなんて，少し前まではまさしく夢のような話だった。

21世紀まであと10年間。夢の世紀だった21世紀も近い将来には現実となる。もっとも，未来は必ずしも明るいわけではないことも確かだ。

最近の世界平和の潮流はひとりの地球人として非常に喜ばしく思っている。しかし，またいつ第2のサダム・フセインが現れないとも限らないのだ。

問題はそれだけじゃあない。人口爆発，そしていまや世界共通の問題である環境破壊。「夢」の21世紀までの最後の10年間である'90年代。いま，僕たちはこれからについて真剣に考えなければならないときにさしかかったのではないだろうか。

Ski

今回は妙に真面目な話でスタートした。とはいっても別に社会問題入門に衣替えしてしまったわけじゃあない。安心してほしい。

問題は雪不足にあるのだ。皆さんご存じのとおり“今年も”雪不足のようだ。これで4年連続ということになる。おまけに昨年は記録的に暖かい1年だった。

これは異常なのではないだろうか。もっとも最近はあまりにも異常ずくめで，何が正常で何が異常なのかが判然としないような気もするけれどね。

とにかく最近は異常が多すぎる。ここまでくると僕みたいな小市民でも地球の将来が心配になってしまう。その思いが冒頭の一節を書かせた，というわけなのだ。

もっとも，これは異常ではないという話もある。なんでもこれは地球のサイクルの一環なんだそうな。

とはいえ，いままで人間が環境に気をつかわなさすぎた，というのも事実だと思う。これからは「地球にやさしい」生活をしようと決意をする筆者であった。

まあこんな決意をさせてしまうほど，雪不足というのはスキーヤーにとっては深刻な問題なのである。とくに僕のようにチームに所属して滑りまくっているスキーマニアには。

そんなフラストレーションを解消すべく，ここで1冊の本を紹介しよう。清水史郎氏による『スキーの科学』という本である。名は体を表すとの言葉どおり，これはスキーに関する本だ。それ以上でも，それ以下でもない。

もっともここで紹介するのだから，スキーだけ，というわけでもない。まあ，あまりじらしてもしょうがないから，そろそろ秘密を明かすことにしよう。

この本はシミュレーションを用いることによってスキーを科学してしまおうというなかなかユニークな発想のもとに書かれている。

ここでシミュレーションと聞いて，何を想像されただろうか。正解はなんとロボットなのである。ロボットとシミュレーション，いったいどんな関係があるのだろう，と思われるかもしれない。確かにいきなりロボットによるシミュレーションと聞いてもピンとこないだろう。

それでは順序立てて説明してみよう。シミュレーションというのは何通りもの方法があるというのは第1回でもお話ししたと思う。前々回，そして前回ではその何種類もあるシミュレーションの中から，コンピュータ上でシミュレーションを実現するために「数学的モデル（数値モデル）」を見てきた。

すなわち，現実を数値に置き換えることによってシミュレートする方法を考えてきた。ところがひと口にシミュレーションといってももうひとつ大きな流れがある。

記憶力のいい方は覚えていらっしゃるかもしれない。「物理的モデル」というのがその名前だ。

「物理」と聞くだけでおそれおののいてしまう文科系の方（もっともそれなら数学もダメか），心配することはありません。ここでいう物理とは「実在のもの」というぐらいの意味なのだから。

つまり，現実を数値に置き換えるのが数学的モデルだとすれば，現実を実在のなにかに置き換えてシミュレーションを行うのが物理的モデルというわけ。

そろそろわかってきたかな。そう，この本では現実のスキーをロボットでシミュレートしているのだ。と聞くといかにもすごそうなロボットを思い浮かべてしまいそうだが，さにあらず。

本を開けてびっくり，おそれおおくも人間様をシミュレートするのは身長30cmの小さなロボット君なのだ。なんでそんな小さなロボットでスキーヤーを真似できるんだろう，そう思いませんか。

ゲレンデでエキスパートの華麗な滑りを見て感動する。スキーヤーなら誰でもが経験することである。ああ，いつかは自分もああなりたい。ボーゲンでこけてばっかりいる現実を忘れ，しばしエキスパートになった自分の滑りを思い描く。

まあ，想像するのは自由ですから。ちなみに，こうしてエキスパートを夢見た人はいつしか3タイプに別れていく。ひとつは夢を追いつづける体育会系タイプ，もうひとつはいつしか上達をあきらめる極楽スキーヤータイプ，そして最後が才能がないのかいつまでたってもこけつづける「ゲレンデ障害物」タイプである。

話がそれたが，上級者の滑りを見ていると，体が実にフレキシブルに動いているのがわかる。雪面を捉えつづける足，ピクともしない上半身，そして華麗にストックを

突く腕，さらに顔はしっかりと前方を見据えている。

よりによって，そんないかにも複雑そうなものをたかだか身長30cmのロボットでシミュレートしようというのだ。嘘だろう，僕だって最初はそう思った。

だが，そのロボットはボーゲン（いちばん基礎的な滑り方）から始まって，パラレル（初心者の憧れの的，これをマスターしていれば一人前だ），さらにはウェーデルン（全スキーヤーの目標，これであなたもエキスパート！）までちゃんとこなしてしまうのだ。まさしくすごいとしかいいようがない。

しかし，これにはちゃんと秘密がある。秘密といっても，もちろんズルをしてるわけじゃない。そこにはすべてのシミュレーションに共通の，成功の秘訣とでもいうべきものがあるのだ。

Secret

さて，その秘訣とはいったいなんなのだろう。それはひと言でいえば「物事をできるだけ簡単に捉えること」ということになる。

これはちょっと変に聞こえるかもしれない。現実をうまくシミュレートするのに，できるだけ簡単にしろだって，というふうに。

普通に考えてみれば，現実をうまくシミュレートするにはできるだけ細かく現実を模倣するほうがいいように思える。確かにこれにも一理はある。

しかし，物事を簡単に捉えることによって初めて見えてくるものも，存在する。

初心者が上級者の滑りを見ると，その全身のすばらしい動きを見て，いったいどこを真似すればいいのかわからなくなってしまう。全身に気をとられるあまり，本質，うまく滑るための本質，を捉えることができなくなってしまうのだ。

ところが，そのうまく滑るための本質「だけ」を真似したロボットだったらどうだろう。これなら本質を見極めることができそうだ。

なにも複雑であることがいい，とは限らないのだ。単純であることの良さというものだってある。なによりも単純なモデルであれば，理解することも，そしてモデルをシミュレートすることも容易なのだから。

さて，そのロボットに教えを乞えば，スキーというのは結局股関節の動かし方いかんなのだ，ということになる。つまり滑りの本質は股関節にあるのだ。それは股関節「だけ」をシミュレートしたロボットが立派にウェーデルンをこなして証明してくれる。

この本はほかにもコンピュータによるスキーのシミュレーションにも少し触れているし，スキー好きの人ならば読んで損はないと思う。

しかし，この本をここでわざわざ取り上げたのは，やはり物事の本質を捉えるというシミュレーションの得意技が見事なまでに証明されているからにほかならない。

ここでお得意の一般化をしてみようか。どうもシミュレーションなどというと，とにかく現実をどこまで細かく真似するか，ということに気をとられてしまいがちだ。

でも，リアルであることがすべてではないのだ，と思う。たとえば，ゲームでジャンボジェットのフライトシミュレータを作ったとする。いくらリアルとはいってもあの何百個もあるスイッチをいじらなきゃならないとすると，プレイする前から興醒めだよね。必要なのは肝心の飛ばす部分なのだ。

要は，シミュレーションもバランスなのだと思う。だいたいの傾向を知りたいのであれば，必ずしもリアルにこだわる必要はないのだ。前回，取り上げた料金所モデルはリアルとはとてもいいがたい。

それでも傾向はつかめるし，どう改善すればいいのかだって知ることができる。スキーロボットだってそうだ。とても人間とは似ても似つかないけれど，本質は捉えている。

よく，シミュレーションをしようとしているのに，あまりにも現実に近づけることにこだわりすぎて肝心のシミュレーションを実現できないでいる人がいる。そんなにリアルにこだわることはないんじゃないのかな，と思うのだ。

逆に，必要であればあくまで現実にこだわることも重要だと思う。必要ならば，だ。たとえば，同じフライトシミュレータでもパイロットの練習用に使うのであれば，やっぱりできるかぎりリアルじゃなきゃ困る（あれ，練習と違うなあ，とかいわれて墜落された日には大変な騒ぎだ）。

Driving

ついでだからもう１冊，別の本にも触れておこう。舘内（タテウチ）端（タダシ）氏の『クルマの速さが見えてきた』という本である。こういう題名だと，おおドラテク（ドライビングテクニック）の本か，と思ってしまう。

よくあるよね，「無敵のドリフトテクニック」とか，「４WD乗りこなしマニュアル」とか（かくいう僕もこの類の本を何冊か持っている）。

しかし，この本はひと味違う。クルマの走りを科学的に分析し，分析を通じていかに速く走るかを考えているのだ。ご存じのように，クルマだって数式で表すことができる。いちばん簡単なのが，例の，

$$F=ma$$

という公式。力は質量に加速度を掛けたものに等しいという，中学で習うアレだ。

この公式からも，同じ力ならば質量が小さいほうが加速度が大きくなる，ということがわかる。言い換えれば，同じエンジンならば車重を軽くしたほうが速くなるということだ。

もっとも，この公式は簡単なほうだ。実際のクルマはもっと複雑なバランスの上で成り立っている。とはいえ，すべてを真似しなくても本質さえ見極めればクルマをうまくシミュレートしてやることができる（先ほども述べたように，重要なのは本質なのだから）。

実際，この本ではコンピュータシミュレーションによって，クルマの「速さ」というものを解析している。コンピュータシミュレーションによる0-400m加速（静止状態から400m走るのにどれだけの時間がかかるのかを競う）や，サーキットでのラップタイムシミュレーションなど，なかなかワクワクするようなことをやっている。

ただ残念なことに，シミュレーションの詳しい方法などまでは書かれていない。一般向けの本だし，スペースの制限もあるだろうから，無理もないとは思うのだが。

しかし，そのシミュレーションの結果だけでも十分に興味深いものがある。いまあるクルマをチューンアップさせる方法として，

1)　軽量化
2)　パワーアップ
3)　タイヤのグリップを上げる

の３つの方法があるとする。これらのうち，どれかひとつによってチューンアップされたクルマでサーキットを走るとすると，どのチューンアップがいちばん効果を発揮するのだろうか。

多くの人が2）のパワーアップと答えるだろう。普通そう思うよね。ところがシミュレーションの結果は，驚くなかれ，3）なのである。

細かいことは実際に読んでもらうとして，

このシミュレーションを裏づけるひとつの事実を述べておこう。

2年前にF1（フォーミュラワン）でターボエンジンが禁止されたことを覚えていらっしゃるだろうか。あの禁止によって，F1のエンジンの出力は1000馬力オーバーからせいぜい700馬力ぐらいへと大幅にダウンしてしまった。

ところが，ところがである。エンジンの出力が大幅に落ちたにもかかわらず，サーキットでのラップタイムはどんどん更新されつづけているのだ。

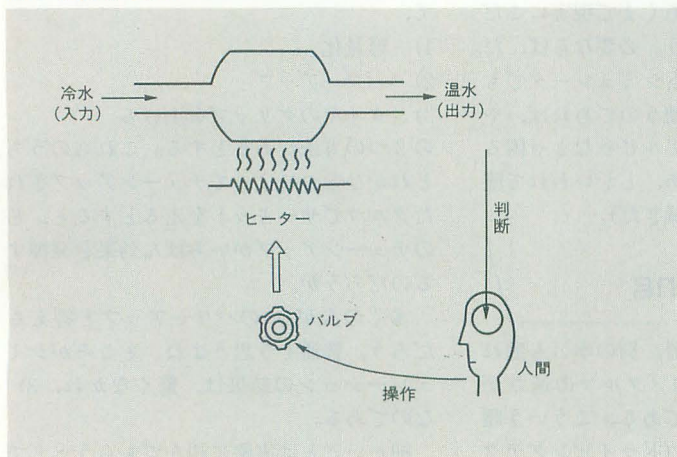
なぜか。これはタイヤの性能によるところが大きい，といわれている。つまり，以前はタイヤがプアなので1000馬力のうちかなりのパワーが無駄になっていたのに対し，いまはタイヤが700馬力のすべてを発揮できるようになった，ということらしい。

もちろん，空力の向上とか，シャーシの進化，ドライバーの腕なんかも関係あるのだけれど。

Continuous

だいぶ前置きが長くなってしまった（それともいままでが本題だったのかも）。今回は連続変化型のシミュレーションに挑戦してみようと思う。

なぜか。早い話が先のサーキットシミュレーションに刺激されたのだ。サーキットシミュレーションをするとなると当然，時間を中心にシミュレートする必要がある。すなわち，連続変化型シミュレーションということになる。

第1回をお読みになった人はご存じだろうが，実は僕には連続型シミュレーションを断念したという忌まわしい過去がある。にもかかわらず，またトライしようというのである。

それだけサーキットシミュレーションに惹かれたってこと。それに，いやしくもコンピュータシミュレーションを語るのであれば，やはり連続型は欠かせないという事情もある。

考えてみれば，最初にシミュレーションをしようとすれば，まずたいていは連続型になるのではないだろうか。現実の世界では（当然）時間は連続なもの，その現実をシミュレートしようとすれば十中八九は連続型で考えはじめることになると思う。

なかには連続型に早々と見切りをつけて，離散型に活路を見いだす僕のような人間もいるだろう。しかし，普通の人間の感覚としてはやはり連続型のほうがなじみやすいだろう。

やはり連続型は避けては通れないようである。もっとも，なじみやすいが難しいというのも事実。だからいきなりサーキットシミュレーションというわけにはいかないだろう（ガクッ，期待させてゴメン）。ここでは比較的簡単なモデルで，連続型シミュレーションの「HOW TO」を見ていくことにしよう。

Heater

モデルとして何を取り上げるか。それがまず大きな問題となる。やはり身近で感覚的に理解しやすいものがいい。

というわけで，ここでは，

「電熱式温水器」

なるものについて考えてみよう。え，電熱式温水器ってなんだ，って。まあ，早い話がシャワーのこと。電気を使って水を温めるシャワーだと思ってくれればいい。図1のようなものだ。

普通僕たちがシャワーを使うとき，どうやって温度調節をしているのだろうか。バルブを開け閉めしてだって。それは当たり前。ここでいいたいのはどのような考え方をして温度調節をしているか，ということだ。熱すぎたらバルブを閉めて，ぬるかったらバルブを開く。おおかたはそれぐらいしか考えていないだろう。僕だってそうだ。しかし，半無意識のうちに，僕たちはかなり高度な制御を加えているのだ。

シミュレーションを通してそれらを明らかにしていこうと思う。

＊　＊　＊

離散型でも連続型でも同じことだけれども，シミュレーションを実現するためにはまず対象となるものの特徴を捉えてモデルを作り上げなければならない。

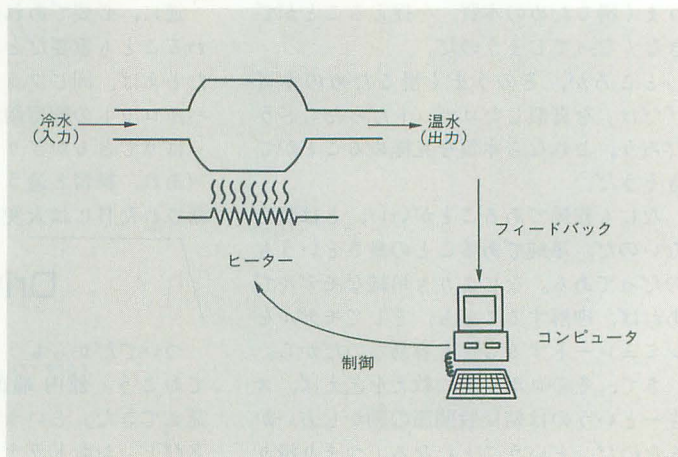
今回の対象である温水器について考えてみると，人間が操作する場合は図1のようになるが，人間の代わりにコンピュータが制御するようになると図2のようになる。このコンピュータによって制御されたシステムをシミュレートすることにしよう。つまり，図2がシミュレーションモデルになるわけだ。

図2のモデルは大きく2つの要素に分けて考えることができる。制御しているほうとされているほうだ。ここではまず制御されているほう（制御対象）について考え，次に制御について考えるということにしよう。

それではまず，制御対象について順を追って考えてみよう。入力に一定の水温の水が流れ込んでくる。この水がヒーターのところまで流れて込んできて，ヒーターによって温められる。そして，その温められた温水がパイプを通って出力に出てくるというわけだ。

というと簡単だが，そうひと筋縄にはいかない。まず第一に，ヒーターで加熱されてもすぐに温水にはなってくれない，とい

図1 温水器モデル

図2 コンピュータ制御

う問題がある。やかんでお湯を沸かすのと同じように，水というのは徐々に温まっていくものなのだから。

専門用語でいうと，これは「1次遅れ」というものである。ま，口で言うより目で見たほうがわかりやすいだろう。リスト1を入力してほしい。今回のプログラムはX1用のみである。もちろん，X-BASICにも来月対応する予定だ。やはり恵まれない環境にあるものを優先すべきでしょう。X68000ユーザーの方，ご了承を。

実行するとまずパラメータの入力画面になるはずである。ならなかったらそれは入力ミスです，はい。パラメータの意味はあとで説明するとして，Kp, Ki, Kdを0に，そしてKcを60に設定してほしい。そして，実行。画面に出たものが1次遅れというものである。

画面は何を意味しているのだろうか。つまりこういうことである。いま，水温が20℃で，目標値は80℃，その差は60℃だ。そこでその60℃の差を埋められるだけの力でヒーターを加熱するのだが，相手は水だ。そうそう簡単には温まってくれない。画面のように，徐々に温まって，目標値80℃に近づいていくのだ。

プログラムを入力しなくてもわかるようにちゃんと図を用意した。図3を見ていただければ理解してもらえると思う。どの図でもそうだけれど，わかりやすくするために多少のデフォルメがある。一応それを頭に入れておいてほしい。

さて問題はこの1次遅れだけではない。ムダ時間というものもある。これはどういうものなのだろうか。とりあえず，ヒーターで水が温められ，温水ができたとする。

図3 1次遅れ系

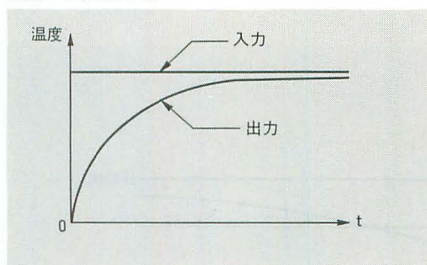
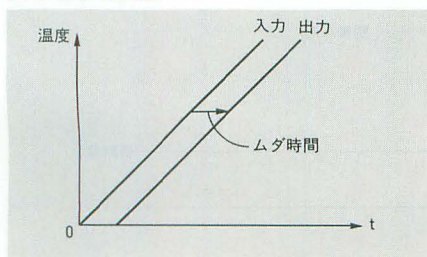



図4 ムダ時間系

しかし，その温水もすぐには出力には出てこない。

なぜなら，ヒーターの出口から出力にたどりつくために一定の時間が必要になるからである。この一定の時間がムダ時間といわれるものである。

これも実際に目で見ていただこう。先ほどプログラムを実行した結果をもう一度見てほしい。パラメータは1次遅れのときと同じ。出力が温まりはじめるまでに，少しタイムラグがあることがわかるだろう。これがムダ時間という代物だ。図4を見てもわかるだろう。

1次遅れとムダ時間，これでとりあえず制御対象の特徴はつかめたことになるわけだ。

wisdom

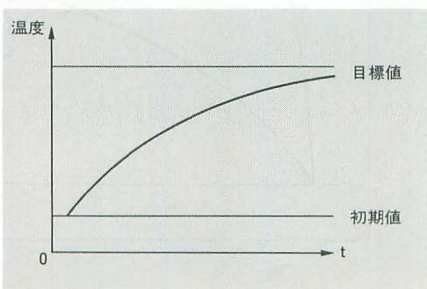
それでは次は，肝心の制御について考えてみることにしよう。

いちばん簡単なのは先ほどやったように，目的の温度に上げられるだけの一定の力をヒーターに加えつづけてやることだ。もっとも，これでは1次遅れとムダ時間によって，図5のように目的温度に達するために随分時間がかかってしまう。

温かいお湯が出てくる前に，風邪でもひきかねない（とはいえ，80℃のシャワーを浴びる馬鹿もいないか）。

そこで「賢い」制御が必要になるわけだ。もっとも，僕たちはその制御を無意識のうちにこなしているのだからたいしたものだ。ここで制御を分析するためには，その無意識にやっていることを明らかにしなければならない。

まず第一に思いつくのは，「現在の値と目標値との差（偏差）に応じてバルブを調節する」というものだ。すなわち，目標値との差が大きければバルブを開き（大パワーで加熱する），差が小さければバルブを閉じる（あまり加熱しない）というものだ。さらに現在値が目標値を上回ってしまった場合（偏差が負になった場合）には，それ

図5 制御なし

に応じて冷やすということになる（おいおい，ヒーターでどうやって冷やすんだ！まあここらへんはモデルということで笑って許してほしい）。

これがいわゆる「比例制御」というものだ。これは理解しやすいだろう。差に応じてコントロールする。当然のなりゆきだ。ところがこの制御には大きな落とし穴が存在する。

お気づきだろうか，この制御では，目標値との差（偏差）がなくなるとまったく加熱しなくなってしまう。すなわち，せっかく目標値に達したものが，また温度が落ちてきてしまうのだ（どんどん水が流れ込んでいるのに加熱しなくなれば，当然，また水温は落ちてくる）。

その結果，決して目標値で安定することはなくなる。それどころか，目標値より低いどこかに安定してしまうのである。つけ加えれば，このときの目標値と安定値の差が永久偏差と呼ばれるものである。

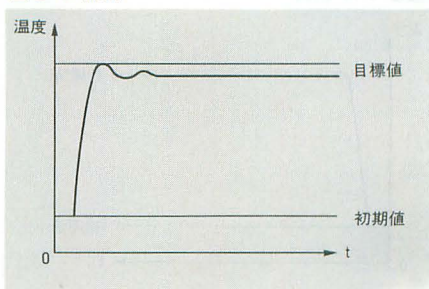
さっそくシミュレーションを見ていただこう。今度はパラメータをKp＝45，Ki＝0，Kd＝0，Kc＝0とセットしてほしい。いったん目標値に達した水温がまた落ちてきてしまうのがおわかりいただけるだろう（図6）。

比例制御ではなにも制御しないのに比べて早く温度が立ち上がるが，永久偏差が生じ，目標値には安定しない。それではどうするか。ここで第2の制御を導入しよう。すなわち，「偏差に応じた速度でバルブを調節する」制御である。

比例制御との違いがわかるだろうか，比例制御は偏差に応じた量でだったのが，この制御では偏差に応じた**速度**に変わっているのだ。

具体的にいえば，偏差が大きければ急いでバルブを開けるが，偏差が小さければ，ゆっくりと落ちついてバルブを開けるということになる。

このような制御を「積分制御」と呼ぶ。なぜ積分なのかはいずれわかるからいいとして，この制御の成果を見てみよう。今度

図6 P制御

はパラメータをKp＝45，Ki＝５，Kd＝０，Kc＝０としてほしい（図７）。

ちゃんと目標値に安定しているのがわかるだろう。これでようやくまともな制御ができるようになったわけだ。

ところが，実はもうひとつ制御方式がある。積分とくれば当然「微分制御」があってもいいだろう。この制御は「偏差の変化速度に応じてバルブを開け閉めする制御」ということになる。

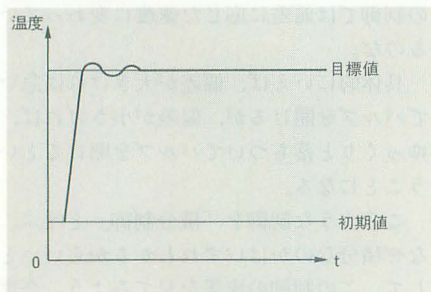
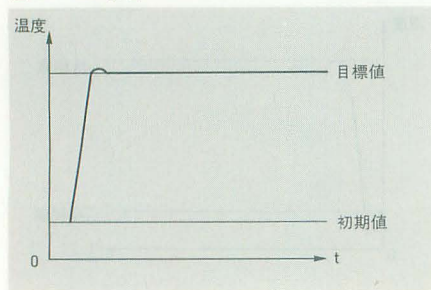
具体的にいうと，現在はまだ目標値より高くても，傾向として水温が下がりつつあるときには，将来下がりすぎることを予想して，バルブを調節するような制御ということになる。

これは将来を予想しながらの制御だから，より高度な制御といえる。この制御がなくても，比例制御と積分制御だけで，目標値に達することはできる。しかし，この制御も併用すれば，それだけ安定した制御が可能になるといえる。その成果を図８に示しておこう。

こうやって３つの制御を見てきたわけだが，

比例（Proportion）
積分（Integral calculus）
微分（Differential calculus）

の３つの単語から，これら３要素すべてを利用した制御をPID制御と呼ぶ。同様にして，比例制御だけの制御をＰ制御，比例制御と積分制御でPI制御ということになる。

ここで制御なしと，Ｐ制御，PI制御，PID制御の比較図を図９にあげておく。こうやって直接比較すると，その違いがよくわかるだろう。

参考までにそれぞれの特徴を述べておこう。

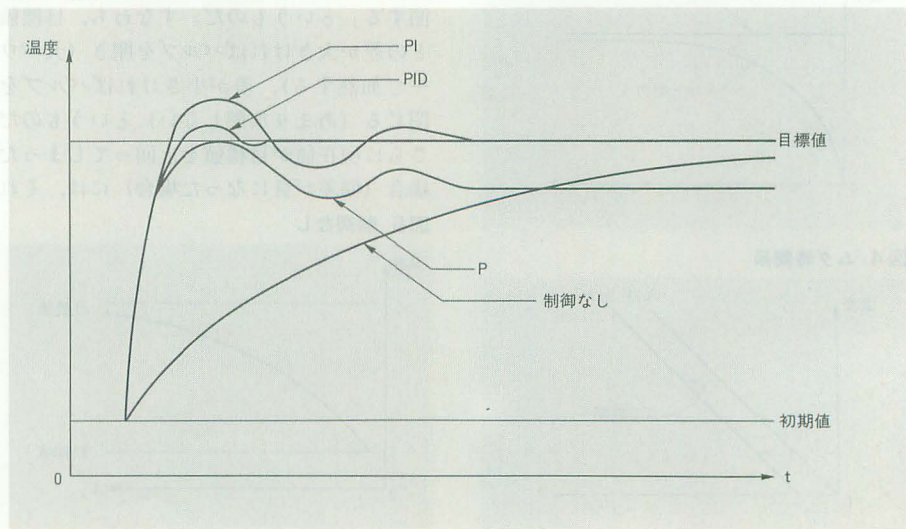
・制御なし
実にゆっくりと目標値へと向かう。
・Ｐ制御
制御なしよりははるかに早く目標値に向かう。しかし目標値には安定しない。
・PI制御
Ｐ制御と同じように早く目標値に向かう。偏差の速度にも応じて制御するため，目標値に安定する。
・PID制御
PI制御と同じように早く目標値に向かう。偏差の将来を見越した制御を行うため，PI制御よりも早く目標値に安定する。

Ｐ，PI，PIDと進むにつれて，より高度な制御になっていっていることがおわかりいただけるだろう。

and……

以上のようにして制御対象と制御の２つの要素を見てきた。これでシミュレーションのモデルを把握できたことになる。

モデルを把握したら，次にやるべきことは当然，モデルをシミュレーションプログラムへと置き換えていくことだ。方向自体は前回までの離散型とあまり変わらないことに気がついたろうか。そう，大筋では離散変化型も連続変化型も同じ道筋を通るのだ。コンピュータ上でシミュレーションを実現するという点でまったく同じなのだから。しかし，どこかで差が出てくるはず。

ずばり，差はこれ以降で出てくる。すなわち，連続変化モデルでは時間を中心にモデルが記述される。たとえばの話，入力をx(t)，出力をy(t)とすると，

$$a \times dy(t)/dt = x(t) - y(t)$$

のようにモデルの状態が時間の関数として表される。

ということは，この数式を解くことによってモデルの状態を知ることができる，ということになる。えっ，この式を解くの，とおそれをなす方もいらっしゃるだろう。確かにこんな微分方程式を解くなんて，楽なことじゃあない。

それじゃあ数学が得意じゃないと駄目か，とあきらめるのはまだ早い。ちゃんとした抜け道があるのである。正攻法が駄目なら，ほかの道を探せということだ。

正攻法がまともに式を解くことだとすると，抜け道というのは，数値的に微分方程式を解いてしまおうという方法だ。詳しい説明は次回に譲るが，この方法ならば，たいていの微分方程式は簡単に解くことができる。微分積分の専門知識など不要。必要なのは本当に簡単な常識だけだ。

というところまできたところで，今回はお開きにしよう。最後に，プログラムのパラメータの説明だけしておこう。

・CLS
０ならば前の画面を消さずに，上書きする。０以外では画面を消去してからシミュレーションを実行する。初期値は１。
・LINE
結果の描画モードを設定する。０ならば点で結果を表示し，０以外であれば線で結果を表示する。初期値は１。
・Target
制御の目標値，初期値は80。
・Step
時間の単位。時刻はStepごとに刻まれる。初期値は0.1であるから，時刻は0.1秒

図７ PI 制御

図８ PID 制御

図９ 制御を比較してみる

ずつ進んでいることになる。

・Kp, Ki, Kd, Kc

制御のパラメータ。0以外の値であれば，その制御が行われる。すなわち制御なしであればKc以外が0，P制御はKp以外が0ということになる。

さらにPI制御ではKd, Kcが0，PID制御ではKcだけが0ということになる。もちろん0以外の場合，値によって制御の効き方が変わってくる。

初期値はそれぞれ45，5，0.5，0。

・k, a

これは次回に説明しよう。もっとも，値をいろいろと変えてみてどんな働きをしているのか調べてみてもいいだろう。

・Xamp, Yamp

結果を表示する際の，X，Y方向の倍率を表す。初期値はそれぞれ100，2。

今回のプログラムでは最初にパラメータをセットした後，シミュレーションを実行する。パラメータは自由に設定できるから，いろいろな値を代入してみて，どんな動きをするのか見ておくといいだろう。

NEXT

今回は，連続変化型シミュレーションについて考えてみた。とはいっても，まだようやくモデルを把握した段階にすぎない。次回はこれをもとにいかにプログラムを作っていくかということを考えていく。

さらに次回は流行のファジィにもふれたいと思っている。なんでも，うまくファジィ制御を行えば，今回のPID制御よりもさらに早く目標値に安定させることができる，というのだ。これをほおっておく手はない。この連載ではシミュレーションに関係するものとあらば，なんでも取り上げていくつもりだ，期待してほしい。

シミュレーションは現実では実現困難なものを仮想の世界で実現させるためのもの，そして未来のことを知るための道具だ。その意味で大いなる可能性を秘めたものであるといえる。未来を知ること，それは人類の永遠の欲望なのかもしれない，と思いつつ，それではまた来月。

参考文献

・「スキーの科学」　清水史郎著　光文社
・「クルマの速さが見えてきた」　舘内端著　光文社
・「ディジタル制御システム」　B.C.クウ著　ホルト・サウンダース・ジャパン
・「ファジィ制御」　菅野道夫著　日刊工業新聞社
・「化学工学実験　1989」　東京大学工学部化学工学学生実験室

リスト1

```
1000 ' Simulation model C1 ver.1.00
1010 '
1020 '  PID Controller for X1 basic
1030 '
1040 '          1990.12 (C) Cammon
1050 '
2000 ' intialize
2010 WIDTH 80,25,0,2: KLIST 0: KMODE 0: CONSOLE 0,25
2020 'width 80                                        'for X1
2030 INIT
2040 DEFDBL d,i,x-z
2050 target = 80
2060 scls = 1: sline = 1
2070 k = .1: a = 1
2080 kp = 45: ki = 5: kd = .5: kc = 0
2090 stp = .1
2100 ampx = 100: ampy = 2
2110 DEF FNx(t) = 20
3000 '
3010 REPEAT
3020  GOSUB 6020
3030  IF scls THEN CLS 0
3040  t=0
3050  idlzo = 0: ido = 0: izo = 0
3060  x = FNx(0)
3070  y1 = x: y2 = y1: y2o = y1: y3 = y2o
3080  do = target - y3: zo = kp*do + kc
3090  mx2 = 0: my12 = 200 - ampy*target
3100  my22 = 200 - ampy*x: my32 = 200 - ampy*y3
3110  IF sline THEN GOSUB 5500 ELSE GOSUB 5020
3500 '
3510 ' main
3520  WHILE ( t*ampx < 639) AND INKEY$ = ""
3530   t = t + stp
3540   x = FNx(t)
3550   y1 = x
3560   y3 = y2o
3570   d = target - y3
3580   id = ido + .5*stp*(do + d)
3590   dd = (d - do) / stp
3600   z = kp*d + ki*id + kd*dd + kc
3610   iz = izo + .5*stp*(zo + z)
3620   dlz = k*(a*iz - (idlzo + .5*stp*dlzo)) / (1 + .5*stp*k)
3630   y2 = dlz + y1
3640   IF sline THEN GOSUB 5500 ELSE GOSUB 5020
3650   ido = id: do = d
3660   izo = iz: zo = z
3670   idlzo = idlzo + .5*stp* (dlzo + dlz): dlzo = dlz
3680   y2o = y2
3690  WEND
3700  LOCATE 42,14: PRINT "Do you need Hardcopy (y/n) n"
3710  LOCATE 69,14: INPUT "",sure$
3720  IF INSTR("Yyﾝ" , LEFT$(sure$,1)) THEN GOSUB 4000
3730  LOCATE 42,14: PRINT "Do you wanna finish the job (y/n) n"
3740  LOCATE 76,14: INPUT "",sure$
3750 UNTIL INSTR("Yyﾝ" , LEFT$(sure$,1))
3760 PRINT CHR$(&HB): END
4000 LOCATE 42,14: PRINT "PID Control Simulator           "
4010 HCOPY 4: RETURN
5000 '
5010 ' drawing results
5020 PSET (t*ampx , 200 - ampy*target , 1)
5030 PSET (t*ampx , 200 - ampy*x , 5)
5040 PSET (t*ampx , 200 - ampy*y3 , 2)
5050 RETURN
5500 ' drawing results - line
5510 mx1 = t*ampx: my11 = 200 - ampy*target
5520 my21 = 200 - ampy*x: my31 = 200 - ampy*y3
5530 LINE (mx2 , my12) - (mx1 , my11) , PSET , 1
5540 LINE (mx2 , my22) - (mx1 , my21) , PSET , 5
5550 LINE (mx2 , my32) - (mx1 , my31) , PSET , 2
5560 mx2 = mx1
5570 my12 = my11: my22 = my21: my32 = my31
5580 RETURN
6000 '
6010 ' set constant
6020 REPEAT
6030  CLS: LOCATE 40,12: PRINT "┌";STRING$(37,"─");"┐"
6040  FOR j=1 TO 11
6050   LOCATE 40 , 12 + j
6060   PRINT "│";STRING$(37," ");"│"
6070  NEXT
6080  LOCATE 40,24: PRINT "└";STRING$(37,"─");"┘";
6090  vr$="CLS": px=42: py=16: cvr=scls: GOSUB 6510: scls=cvr
6100  vr$="LINE": px=60:py=16: cvr=sline: GOSUB 6510: sline=cvr
6110  vr$="Target":px=42:py=17:cvr=target:GOSUB 6510:target=cvr
6120  vr$="Step": px=42: py=18: cvr=stp: GOSUB 6510: stp=cvr
6130  vr$="Kp": px=42: py=19: cvr=kp: GOSUB 6510: kp=cvr
6140  vr$="Ki": px=60: py=19: cvr=ki: GOSUB 6510: ki=cvr
6150  vr$="Kd": px=42: py=20: cvr=kd: GOSUB 6510: kd=cvr
6160  vr$="Kc": px=60: py=20: cvr=kc: GOSUB 6510: kc=cvr
6170  vr$="k": px=42: py=21: cvr=k: GOSUB 6510: k=cvr
6180  vr$="a": px=60: py=21: cvr=a: GOSUB 6510: a=cvr
6190  vr$="Xamp": px=42: py=22: cvr=ampx: GOSUB 6510: ampx=cvr
6200  vr$="Yamp": px=60: py=22: cvr=ampy: GOSUB 6510: ampy=cvr
6210  LOCATE 42,14: PRINT "Are you sure (y/n) y"
6220  LOCATE 61,14: INPUT "",sure$
6230 UNTIL INSTR("Yyﾝ",LEFT$(sure$,1))
6240 RETURN
6500 '
6510 LOCATE 42,14: PRINT LEFT$(vr$+"      ",6)+" : ";cvr
6520 LOCATE 51,14: INPUT "",cvr
6530 COLOR 4:LOCATE px,py:PRINT LEFT$(vr$+"      ",6)+" : ";cvr;
6540 COLOR 7: LOCATE 42,14: PRINT STRING$(36," ")
6550 RETURN
```

ようこそここへC言語

[第4回]

配列って何だろう(その1)

Nakamori Akira
中森 章

プログラミング言語を学ぶ人が，初めて具体的なデータ構造というものに出会うのがこの配列でしょう。といっても難しい概念ではありません。ここでは，背景となる考え方から学んでいきましょう。今回は一次元の配列です。

魅力的なソフトがないのでスーパーファミコンを買うのはよそうと思っていた矢先，ドラクエVの発表があってあっさりと心変わりした中森章です。それにしても，発売されたとはいえ，おもちゃ屋の店頭で見つけることのできないスーパーファミコンが手に入るのはいつのことなのでしょう。

さて，今回のテーマは配列です。配列はすべてのプログラミングにおいて基本になるデータ構造です。今回は配列のうちでももっとも基本的な一次元配列について学んでいきましょう。なお，前回の予告では今月のテーマが配列と文字列ということになっていましたが，予定を変更して今回は配列だけです。文字列については今回と次回で配列に関しての知識を深めたあとの次々回に説明したいと思っています。

配列とは何か

以前も説明しましたが，プログラムの基本はある変数の値を加工して別の変数に格納することです。プログラムではいろいろなデータ型の変数を扱いますが，変数の中には本質的に同一のものが存在しています。それらの変数をまとめて統一的に扱うためのデータ構造が配列です。

例として，何人かの人が受けた試験の得点を格納する変数を考えましょう。単純に考えれば，1番目の人の得点を格納する変数はmark1，2番目の人の得点を格納する変数はmark2，3番目の人の得点を格納する変数はmark3，……というように別々に変数を宣言していきます。ここで，

mark1, mark2, mark3, ……

という変数はある試験の得点という意味で同一のものです。また，そのデータ型も同一のはずですね。今，試験を受けた人が4人いるとして，試験の得点の合計を求めることを考えましょう。合計値を格納する変数をsumとすると，sumを求めるためのプログラム(式)は，

sum＝mark1＋mark2＋mark3＋mark4

となります。この場合，人数が4人ですから合計値をすっきりとした式で表すことができますが，これが1000人だった場合はどうでしょう。まず変数の宣言が大変です。C言語での変数宣言を考えると，

int mark1, mark2,……, mark999, mark1000；

となり，とてつもない文字数です。また合計値を求める場合も，

sum＝mark1＋mark2＋mark3＋
　……＋mark999＋mark1000；

というように変数宣言と同じくらい(少し多い)の文字数になってしまいます。こんな大変な思いをしてまでプログラムを書く気は起こりませんね。そこで登場するのが配列というわけです。配列とは，日常語と同じく，

順序を決めた並べ方(並べたもの)

という意味です。そして，その考え方は，

変数を順序だけで参照して扱う

ということになります。先の得点の例でいえば，

mark1　は　1番目のデータ
mark2　は　2番目のデータ
…………
mark1000　は　1000番目のデータ

として扱うのが配列の考え方です。これは変数を番号で間接的に参照することにほかなりません。番号で参照するということは，その番号として整数型変数の値を使用できるということです。番号を変数で指定できるとなると，先の例の得点の合計値は，前回学んだ繰り返し制御構造を使って，

```
sum=0;
for(i=1; i<1000; i++)
    sum=sum+ [ i番目のデータ ] ;
```

と，いとも簡単に記述できてしまいます。そしてこのこと，すなわち，大量のデータを簡潔に扱うことこそが配列というデータ構造が存在する意味なのです。

C言語で配列を扱う

それでは，C言語における配列の使用法を具体的に説明しましょう。C言語で配列を扱うためには，

- 配列の宣言
- 配列要素の参照
- 配列の用途

を押さえておけば十分です。以下に順次説明していきましょう。

●配列の宣言

配列とはデータをいくつか並べたものですから，配列を宣言するためにはデータ(配列要素)のデータ型とデータの個数(要素数)を決定することが必要です。これらが決まると配列は，

データ型　名前［定数式］；

という形式で宣言することができます[1]。［と］で囲まれ

PROGRAMMING

た定数式が配列の要素の個数を指定している部分です。さらに,

名前［定数式］

の部分をカンマ「,」で区切って並べることで，同じデータ型の要素を持つ複数の配列を一度に宣言することも可能です。ところで，C言語では～型の配列という言葉をよく聞きますが，これは配列要素が～型の配列という意味です。

配列宣言の例を示しましょう。1000個の単精度浮動小数点データ（float型）を配列要素とする配列fpaの宣言は,

float fpa［1000］;

となります。また，配列fpaと同時に10個の単精度浮動小数点データを配列要素とする配列fpbを宣言する場合は,

float fpa［1000］, fpb［10］;

となります。

これらの例を見てわかるように，配列の宣言は配列名に［ ］が付随しているほかは変数の宣言とまったく同じです。実際，配列の宣言とただの変数の宣言を,

float fpa［1000］, fe, fpb［10］;

というようにカンマで区切って同時に行うこともできるようになっています（feがただの変数です）。結局，配列の宣言も変数の宣言の一種なのです。この意味で，当然のことながら，配列の宣言はプログラム中で変数の宣言をしている部分と同じ場所に書かなければなりません。

[1] ここではもっとも単純な場合の宣言を示してある。C言語ではポインタの配列や関数の配列といった宣言も可能であるが，これらについては今後の連載で説明する予定である。

●配列要素の参照

配列を宣言するということは配列の要素を番号で参照するということです。この考え方はC言語に限らずすべての（配列型を持つ）プログラミング言語で共通ですが，番号の付け方は各プログラミング言語で必ずしも同じではないので注意が必要です。C言語では配列要素を参照するための番号（これを添字という）は0から始まります。すなわち，1番目の要素を参照するときの添字は0で，最後の要素を参照するときの添字は（要素数－1）です。

それでは，配列要素の参照方法です。C言語では配列要素を参照するために専用の演算子が用意されています。この演算子を使用して，配列名と添字を演算することで，添字で指定する配列要素を参照することができるようになっています。少し難しい表現をしてしまいましたが，その内容はぜんぜん難しくありません。C言語で配列要素を参照する演算子とは［］のことなのです。どこかで見たことはありませんか。そうです，配列を宣言するときに要素数を指定する［と］は演算子としても使用されるのです[2]。

このとき［］内には添字（を表す式）を指定し，それを配列名のあとに書きます。たとえば,

fpa［2］

という表現は配列fpaの添字2（つまり3番目の）の要素を示しています。もし，この配列要素の値を1にしたいのであれば,

fpa［2］=1;

という式を書けばいいですし，逆にこの配列要素の値を別の変数xに代入したいのであれば,

x=fpa［2］;

という式を書けばいいでしょう。演算子などと難しいことをいっていますが，ごく自然な方法で配列要素を参照できるのがわかるでしょう。

配列要素の参照の方法についてくどくどと説明してきましたが，実用上は,

float fpa［1000］;

という1000個の要素を持つ配列の宣言は,

fpa［0］, fpa［1］, fpa［2］,……, fpa［999］;

という添字で参照できる1000個の変数を宣言したのだと思って差し支えないでしょう。

[2] 配列宣言時の［ ］は演算子ではなく，単なる記号である。まぎらわしいといえばまぎらわしい。逆に，FORTRANやPASCALなどほかのプログラミング言語からの類似（宣言時に個数を指定する記号と参照に使う記号が同じであること）と思えば当然である。

●配列の用途

配列の宣言と参照方法を押さえたところで，配列の用途について考えましょう。

配列のイメージは番号（添字ですね）を付けられた変数ですから，容易に考えられる用途は統計処理です。ちょっと表現が大袈裟ですが，要は複数個のデータを番号を付けて扱うことができるということです（先に述べた試験の得点の合計を求める例もこれですね）。一連のデータを添字付きで扱うことができれば，総和，平均，標準偏差など統計量の計算は思いのままです。また，昇順，降順のソーティング（並び換え）にも便利です。

配列のもうひとつの用途はテーブル，つまり表の実現です。この場合，配列の添字は順番を示す単なる数字でなくなんらかの意味を持っています。たとえば，配列の添字が整数の集合を代表している場合を考えます。このとき，ある整数値（これが添字になる）に対応する配列の値は，添字である整数になんらかの変換を加えた結果（難しい言葉でいえば写像）とみなすことができます。

具体例を示すと,

root［0］ が 0.0
root［1］ が 1.0
root［2］ が 1.414
root［3］ が 1.732
……

であるような浮動小数点型の配列rootは添字で与えられる整数の平方根を値としています。これは平方根の表にほかなりません。また，添字を整数以外の項目に対応させることもできます。たとえば，添字0の値は身長，添字1の値は体重，添字2の値はバスト，添字3の値はウエスト，添字4の値はヒップというように考えることで，配列をある個人に関するデータを示す表とみなすこともできるのです[3]。

配列の用途を大ざっぱに分類すれば上の2つになります[4]。2つの違いはプログラムで配列の要素や要素の順

序を変更するかしないかによります（要素や順序を変更しないのが表です）。しかし，これはこういう使い方が多いというだけで配列を用いるプログラムの書き方になんら制限を課するものではありません。

実際のプログラムでは配列を添字付き変数として使用しているのか表として使用しているのか区別できない場合のほうが多いかもしれません。また，配列の要素に関しても，それがさらに配列であったり，構造を持った複合的なデータ型だったりして複雑な使われ方をしていることが結構あるようです。それらは，見掛けは複雑です（それだけいろいろなことに使えるということ）が，基本的な考え方は上の2つの用途から大きくはずれるものではありません。私たちの心づもりとしては，

・いくつかの変数を添字を付けて統一的に扱う
・表を実現する

という必要があるときには配列を使用するということです。

---

3) 身長，体重，……など，ある1つのデータに関する異なる属性をまとめて表現する場合は構造体を使うのが一般的。

4) 実は，配列にはもう1つの用途がある。C言語では文字(char型)の配列で文字列を表す。この場合，配列の要素が変わることも要素の並び方が変わることもないので一種の表と見なせないこともない。文字列についてはのちの連載で説明する予定である。

配列の初期化

配列を使用する場合，特に配列を表として使用する場合，配列要素にあらかじめ初期値を与えておきたいことがあります。ここでは配列の要素を初期化する方法について説明しておきましょう。

配列の初期化は配列を宣言するときと同時に行います。このとき，宣言のあとに配列要素カンマ「,」で区切った初期値のリストを{と}で囲み，=でつなぎます[5]。たとえば，添字が月の番号（1月，2月という）を表し，配列要素が添字で与えられる月の日数（閏年は考えない）を値とするようなint型配列daysを初期化するためには，その宣言時に，

```
int days [13] = { 0, 31, 28, 31, 30, 31, 30,
                  31, 31, 30, 31, 30, 31};
```

と書けばよいのです。ここで配列の要素数が13個になっているのは添字の上限を12にし，添字nがn月に対応するようにするためです。添字0の部分はダミーで使用しません。

配列の初期化においては，配列名から初期化の終わりの}までがひとつの変数（配列）の宣言と見なせます。したがって初期化付きの複数の配列の宣言をカンマで区切ってほかの変数宣言と同時に行うこともできます。

```
int a [3] = {1,2,3}, b, c [5], d [2] = {100,200};
```

という変数宣言の記述が何を意味するかは説明するまでもありませんね。

ところで，C言語では要素数を指定しない配列宣言も許されます。これを不完全型（大きさが不定という意味）の配列宣言と呼びます。不完全型の配列を使用するためにはなんらかの方法で完全型にしなければなりませんが，ここでは難しい話はやめておきましょう。不完全型の配列を完全型にするためのひとつの方法が初期化であるということを覚えておいてください。そのとき，初期化リストの中の要素の個数が配列の要素数と見なされます。

```
int a [] = {1, 2, 3, 4};
```

という不完全型の配列の初期化では，配列aの要素数は4個と見なされ，その要素であるa [0]，a [1]，a [2]，a [3] の値をそれぞれ1,2,3,4に初期化します。この（初期化を付けた）不完全型の配列宣言は要素数が非常に多い表を配列で実現するとき，あるいは要素数があとから変更される可能性のある表を配列で実現するときによく使用されます。

話は横に逸れますが，このような配列の要素数を計算するためにはsizeofという演算子を使えば簡単です。sizeofとは変数の占める領域の大きさ（バイト数）を計算する演算子ですから，

```
sizeof(配列名)/sizeof(配列要素のデータ型)
```

が配列の要素数になります。たとえば，

```
float fpa [ ] = {1.0, 1.4, 1.7,……};
```

と宣言されている配列の要素数は，

```
sizeof(fpa)/sizeof(float)
```

という式で知ることができます。

---

5) 配列を初期化する要素の形式は定数式（コンパイル時に値が計算できるもの）が基本である。関数の先頭で行われる（動的な）配列の初期化ではその実行時まで値の決まらない式でも許される。つまり，

```
f(x)
int x;
{    int a [3] = {x+1,x+2,x+3};
     ........
}
```

のような初期化も可能である。x+1, x+2, x+3という値は関数fが呼ばれたときに計算され配列aに代入される。このような初期化はGCCではもちろん可能であるが，XCでは制限が付く。XCのver.1.0では動的な配列宣言を初期化することはできない。XCのver.2.0では動的な配列宣言の初期化は定数式のみが許される。

◆基礎力を高めよう

設問1　次に示す配列の宣言が（文法的に）正しいときは○，誤っているときは×をつけてください。

1) int x [ ] = {1, 2, 3,};
2) int y [2] = {1, 2, 3};
3) int z [ ] = {2,};
4) int w [3] = {1};

5) int v [3] = {1,};
6) int u [2] = {1+2, 3＊4/2};
7) int a [2＊3+1] = {0};
8) int b [3] = { };
9) int c [ ] = { };
10) int d [ ];

設問２ 次に示す配列要素の参照方法が（文法的に）正しいときは○，誤っているときは×をつけてください。ただし，

int x;
int array [3] = {1,2,3};

がすでに宣言されているものとします。

1) x = array [2];
2) x = array [1]+2;
3) x = array [−2];
4) x = 2 [array];
5) x = [array] 2;
6) x = [2] array;
7) array·[2] = x;
8) 2 [array] = x;
9) array [1]+2 = x;
10) x = (x+1) [array];

（解答は68ページ）

一次元配列を扱うプログラム

それでは，配列を使った実際のプログラムを紹介しましょう。ここでは典型的な配列の用途に従って２つのプログラムを作っています。すなわち，統計処理（もどき）のプログラムと配列を表として使うプログラムです。

●統計処理もどき

与えられた配列に何人かの試験の得点が格納されている場合に，得点の平均と標準偏差を計算し，さらに各自の得点の偏差値を計算するプログラムがリスト１です。得点はあらかじめmarkというint型配列の中に初期値として格納されています。

このとき，平均や標準偏差は次の式で計算できます（nは要素数）。

平均 = (mark [0]+mark [1]+……+mark [n])/n

分散 = {(mark [0]−平均)^2 +(mark [1]−平均)^2 +……

……+(mark [n]−平均)^2 }/n

標準偏差 = √分散

また，受験生にとって頭の痛い偏差値は，

(mark [i]−平均) ＊10/標準偏差 + 50

という式で計算することが多いようです（この式は添字iの得点の偏差値）。これだけの式がわかれば，あとは繰り返し制御構造であるforループを使って簡単に計算できますね。

ところで，リスト１のプログラムはint型とdouble型の変数をそのまま演算しています。このような場合，Ｃコンパイラが（もっとも自然な結果が出るように）適当に変数の型を変換して計算してくれるのです[6]。このことも覚えておきましょう。

●配列を表として使う

いくつかの値に対する三角関数（正弦）の値を格納してある配列を使って，入力された任意の値に対する正弦値を計算するプログラムがリスト２です。リスト２のsinTabという配列には三角関数sinへの入力値を０から π/4の範囲で0.01きざみに変化させたときの値が入っています。すなわち，

sinTab [0] は sin(0)
sinTab [1] は sin(0.01)
sinTab [2] は sin(0.02)
sinTab [3] は sin(0.03)
…………

リスト１

```
 1: /*
 2:          配列で統計的な処理を行う
 3:
 4:          配列 mark に格納されている
 5:          得点の平均と標準偏差を求め、
 6:          各自の偏差値を計算する
 7: */
 8:
 9: /* 平方根を求める関数 sqrt を使うためのオマジナイ */
10: #include <math.h>
11:
12: int mark[]={
13:  60, 90, 15, 81, 65, 100, 75, 80, 85, 72, 55, 60, 85,
14:  77, 10, 48
15: };
16:
17: int v[sizeof(mark)/sizeof(int)]; /* 偏差値を入れる */
18:
19: double heikin,  /* 平均     */
20:        hyojun,  /* 標準偏差 */
21:        bunsan,  /* 分散     */
22:        sum;     /* 合計     */
23:
24: int    n=sizeof(mark)/sizeof(int); /* 要素数   */
25:
26: main()
27: {
28:         int i;
29:
30:         sum=0;
31:         for(i=0;i<n;i++)
32:             sum = sum + mark[i];
33:         heikin = sum/n; /* 平均を求める */
34:
35:         sum=0;
36:         for(i=0;i<n;i++)
37:             sum = sum + (mark[i]-heikin)*(mark[i]-heikin);
38:         bunsan = sum/n;          /* 分散というやつ */
39:         hyojun = sqrt(bunsan);   /* これが標準偏差 */
40:
41:         for(i=0;i<n;i++)         /* 偏差値の計算です */
42:             v[i] = (mark[i]-heikin)*10/hyojun + 50;
43:
44:         for(i=0;i<n;i++)
45:             printf("番号 %d¥t: 得点 %d¥t: 偏差値 %d¥n",
46:                           i,       mark[i],      v[i] );
47:         printf("平均 %f : 標準偏差 %f¥n", heikin, hyojun );
48:
49: }
```

リスト１の実行結果

```
番号 0  : 得点 60    : 偏差値 47
番号 1  : 得点 90    : 偏差値 59
番号 2  : 得点 15    : 偏差値 28
番号 3  : 得点 81    : 偏差値 56
番号 4  : 得点 65    : 偏差値 49
番号 5  : 得点 100   : 偏差値 63
番号 6  : 得点 75    : 偏差値 53
番号 7  : 得点 80    : 偏差値 55
番号 8  : 得点 85    : 偏差値 57
番号 9  : 得点 72    : 偏差値 52
番号 10 : 得点 55    : 偏差値 45
番号 11 : 得点 60    : 偏差値 47
番号 12 : 得点 85    : 偏差値 57
番号 13 : 得点 77    : 偏差値 54
番号 14 : 得点 10    : 偏差値 26
番号 15 : 得点 48    : 偏差値 42
平均 66.125000 : 標準偏差 24.196784
```

という関係になっています。したがってsinを求めるべき値が与えられると，それを100倍した値(端数は切り捨て)を添字と見なして配列sinTabを参照すれば値が得られるという仕組みです。このやり方は入力が，

0.xx

というように小数点以下2位までで表せる場合はいいのですが，もっと半端な場合も計算できなければおもしろくありません。そこで入力を100倍した値がi（端数を切り捨てている）であるとき，

sinTab［i］からsinTab［i+1］までの値が
入力に対して直線的に変化している

と仮定して入力の端数で直線近似を行うことにします。

リスト2ではsinTab［i］とsinTab［i+1］の間を10000分割して近似値を求めています。リスト2では実際のsin関数の値と比較するようになっていますが，結果は有効数字4桁程度の精度で正しく計算できているようです。思ったほどの精度は出ませんでしたが，まあ愛敬ですね。

リスト2

```
 1: /*
 2:             配列を表として使う
 3:
 4:             三角関数 (sin) を近似計算する
 5: */
 6:
 7: /* 三角関数 sin （答え合わせ用）を使うためのオマジナイ */
 8: #include <math.h>
 9:
10: /* この配列には sin の値を0からπ/4の範囲で0.01きざみに格納 */
11: double sinTab[]={
12:     0.000000000000000,   0.009999833334166,   0.019998666693333,
13:     0.029995500202496,   0.039989334186634,   0.049979169270678,
14:     0.059964006479445,   0.069942847337533,   0.079914693969173,
15:     0.089878549198011,   0.099833416646828,   0.109778300837170,
16:     0.119712207288920,   0.129634142619690,   0.139543114644240,
17:     0.149438132473600,   0.159318206614250,   0.169182349067000,
18:     0.179029573425820,   0.188858894976500,   0.198669330795060,
19:     0.208459899846100,   0.218229623080870,   0.227977523535190,
20:     0.237702626427130,   0.247403959254520,   0.257080551892160,
21:     0.266731436688830,   0.276355648564110,   0.285952225104840,
22:     0.295520206661340,   0.305058636443440,   0.314566560616120,
23:     0.324043028394870,   0.333487092140810,   0.342897807455450,
24:     0.352274233275090,   0.361615431964960,   0.370920469412980,
25:     0.380188415123160,   0.389418342308650,   0.398609327984420,
26:     0.407760453059570,   0.416870802429210,   0.425939465066000,
27:     0.434965534111230,   0.443948106965520,   0.452886285379070,
28:     0.461779175541480,   0.470625888171160,   0.479425538604200,
29:     0.488177246882910,   0.496880137843740,   0.505533341204850,
30:     0.514135991653110,   0.522687228930660,   0.531186197920880,
31:     0.539632048733970,   0.548023936791870,   0.556361022912780,
32:     0.564642473395040,   0.572867460100480,   0.581035160537310,
33:     0.589144757942270,   0.597195441362390,   0.605186405736040,
34:     0.613116851973430,   0.620985987036560,   0.628793024018470,
35:     0.636537182221970,   0.644217687237690,   0.651833771021540,
36:     0.659384671971470,   0.666869635003700,   0.674287911628150,
37:     0.681638760023330,   0.688921445110550,   0.696135238627360,
38:     0.703279419200410
39: };
40:
41: int     n=sizeof(sinTab)/sizeof(double); /* 配列の要素数 */
42: double x,y;
43: int     i,j;
44:
45: main()
46: {
47:         printf("浮動小数点(約0.785以下)? ");
48:         scanf("%lf",&x);
49:         i=x*100;                    /* 100倍すれば添字になる     */
50:         j=x*1000000-i*10000;        /* 小数点第3位以下を求める */
51:
52:         y=( sinTab[i+1]*j + sinTab[i]*(10000-j) )/10000;
53:                 /* 小数点第3位以下に値があるときは直線近似 */
54:
55:         if(i>=(n-1)){
56:                 printf("%f は範囲外です¥n", x);
57:         }
58:         else{
59:                 printf("sin(%f):%f ≦ %f ≦ %f (%f)¥n",
60:                         x, sinTab[i], y, sinTab[i+1], sin(x));
61:         }
62: }
```

リスト2の実行結果

```
浮動小数点(約0.785以下)?
sin(0.555555):0.522687 ≦ 0.527408 ≦ 0.531186 (0.527414)

浮動小数点(約0.785以下)?
sin(0.234567):0.227977 ≦ 0.232418 ≦ 0.237702 (0.232422)
```

---

[6] 異なる型どうしの演算ではちゃんと型変換を明示したほうがいいという意見もある。型変換は変数の前に変換するデータ型の名前を（ ）で囲んで示す。たとえばvarがint型の変数であるとき，

(double)var

はdouble型への型変換になる。これを型のキャストという。

＊

今回はC言語のデータ構造の中でもっともポピュラーな一次元配列の解説をしました。配列は添字でデータを扱えるのが最大の特徴です。前回説明した繰り返し制御構造はまさにこの配列要素を扱うのに最適な制御構造です。制御構造と配列を征服すれば，ちょっとしたプログラムならば書けるようになるはずです。これらはC言語のプログラミングの基本ですから，各自大いに復習，自習に励んでくださいね。来月は一次元以上の配列について説明したいと思います。それではまたお会いしましょう。

＊　＊　＊

◆基礎力を高めようの解答

設問1　1)○　2)×　3)○　4)○　5)○
6)○　7)○　8)○　9)○　10)○

解説

1)，3)，5)：初期化リストの最後のカンマはあってもなくてもよい。UNIX上のコンパイラでは最後に余計なカンマを付けるとエラーになることもある。

4)，5)，7)：初期化リストの要素数が配列の要素数に満たないときは，残りの要素は0で初期化される。……ことになっているが，GCCやXCでは残りの要素の値はなぜか不定になっている。

6)，7)：配列の要素の個数の宣言や初期化リスト内の初期値は結果が定数になる式で書いてよい。

2)：初期化リスト内の要素数は宣言した配列の要素数より多くてはいけない。XCではエラーになるが，GCCでは警告に留まり配列の要素数を優先する。

8)：初期化がない配列宣言と同じこと。

9)：文法的には正しいが意味的にはおかしい。初期化を行うということは(この場合は実際には初期化を行わないが)，そこに実体があるということである。しかし，配列宣言自体は不完全型であり大きさが不定になっている。これは実体のない配列の宣言したことになる。このような配列の要素を参照しても意味がない。10)との違いに注意すること。

10)：これはdという変数がint型の配列であるということを示しているだけで，dという配列を宣言したわけではない。本当の配列dの宣言は（おそらく）別ファイルの中で行われている。

＊　＊　＊

設問2　1)○　2)○　3)○　4)○　5)×
6)×　7)○　8)○　9)×　10)○

解説

4)，8)，10)：配列要素を参照する演算子は，

x［y］

の形でのみ使用できる。通常はxが配列名，yが添字であるが，xとyが逆でもかまわない。したがって，4)，8)，10)のような記述もできる。ただし，このような記法をする人はよほどのへそまがりである。

3)：［ ］は単なる演算子であるから添字として与える値は負の数でもよい。負の添字は配列要素の参照では意味がない（許される添字の範囲外）が，ポインタに対して［ ］を用いるときにはそれなりの意味がある。ポインタと配列の関係はこの連載でもそのうちじっくりとやりたいと思うが，今はこの程度の理解でよい。

9)：代入文の左辺に式が書けないのは変数の場合と同じ。

ソーティングプログラム(前編)

Murata Toshiyuki 村田 敏幸

プログラミングを学ぶ者が一度は心ひかれるソートの技法。アルゴリズムの違いが大きな効果となって表れるだけに，多くの先人たちがさまざまな手法を研究してきました。アルゴリズムを考える楽しさと実際にコーディングする際の工夫を味わってください。

今回はだいぶ前に予告しておいた“ソート”を取り上げる。予想外に原稿が膨れ上がってしまったので（失敗），ブチンと2つに割って，前後編の2回に分けてお届けする（ま，のんびりやるさ）。

ソート（sort)。名詞形ならソーティング（sorting)。並べかえ，整列，分類，順序づけ，などと訳される。説明するまでもなく，データ列を決められた順序に並べかえる操作のことをいう。ソートは，サーチ（search：探索，検索）と並んでアルゴリズムの良否が露骨に実行時間に反映する操作だ。だからなのだろう，とっくの昔に諸先生によって研究し尽くされて，アルゴリズムが確立している分野でもある。今回は，そうやって先達が考え出したアルゴリズムを棚からボタモチとばかりに拝借して，68000のアセンブリ言語でコーディングしてみようというわけだ。

参考文献としては，Wirthの名著『Algorithms+Data Structures＝Programs』，あるいは，その改訂版である『Algorithms&Data Stuructures』を勧めておく（邦題はそれぞれ『アルゴリズム＋データ構造＝プログラム』，『アルゴリズムとデータ構造』)。より深く知りたい人にはKnuthの『The Art of Computer Programming』のうちの1冊(邦題は何ていったかな？ 失念した）がある。

準備

今回は，各ソートアルゴリズムごとに，

・アルゴリズムの概要

・プログラム例

・改良案（アルゴリズム上の改良だったり，マイナーな最適化だったり，いろいろ）

を，たんたんと並べるスタイルで進める。プログラム例はソートを行うサブルーチン本体のみとし，各サブルーチンは以下の仕様に統一した。

・扱うのは32ビット符号つき数値の配列。

・ソートは昇順[1]に行う。

・引数は“配列の末尾のアドレス＋1”と“配列の先頭アドレス”をこの順でスタックに積んで渡す。

・サブルーチン名は常にsort。

●動作試験用プログラム

今回の各リストでは，冒頭に必ず注釈として，8000個のランダムデータからなる配列をソートしたときの実測時間[2]を1/100秒単位で入れてある。ソート時間の計測には，リスト1に示すプログラムを使った。脇役にすぎないプログラムだが，いちおう簡単に触れておく。

リスト1は，IOCSコールONTIMEを使って1/100秒単位でソート時間を計り，表示する。ONTIMEはX68000を起動してからの時間を1/100秒単位で返すから，ソートルーチンをコールする直前と直後にONTIMEを呼べば，その戻り値の差でソートにかかった時間が得られる。このような計時のときには，計時開始時と終了時とでそれぞれ単位時間（いまの場合は1/100秒）の誤差が発生する可能性がある。リスト1では，28～32行で“1/100秒単位の時間の変わり目”を待つことで,計時開始時の誤差は（ほとんど）消しているが，終了時にはなにもしていない。リスト1では，切り捨てを行う方向に，まだ最大1/100秒の誤差が残っていることを覚えておいてもらいたい[3]。

1/100秒単位で得た時間を文字列に変換する処理には，手抜きして，FLOATn.Xのファンクションコール USINGを使っている（というわけで，テストプログラムの実行時にはFLOATn.Xが組み込まれていなければならない)。FLOATn.Xのファンクションコールについては，きっと，そのうち“算術計算(!?)”かなんかのテーマのときに取り上げる機会があると思うから，いまは無視してしまう。どうしても気になる人は XC Ver.1.0（Ver.2.0は不可）のFEFUNC.Hというファイル中の注釈か，数値演算プロセッサボードのマニュアルか，市販の解析本『X68000環境ハンドブック』を参照されたい。

●動作試験の方法

試験の対象となるソートルーチンとリスト1とは，リンク時に結合する。リンクは（ソートルーチンのオブジェクトファイル名がSORT1.Oだとすると)，

A＞LK SORTTEST SORT1

1) 昇順とは小さい順のこと。逆は降順。

2) テストは，OPMDRV.XはOFFにし，そのほか割り込みをかけまくる常駐プログラム・デバイスドライバの類もはずした状態で行った。

3) もっとも，誤差は計測の常ではある。なお，本当は30～33行のような細工をせずに，数回実行して平均をとったほうが，より正確な時間が得られるはずだが，今回はその手間を惜しんだ。

とやってもいいが，

A>LK SORTTEST SORT1/oSORT1.X

あるいは，

A>LK SORT1 SORTTEST[4]

のようにして，ソートルーチンごとに実行ファイルにも別のファイル名をつけたほうがなにかと楽だ。

テスト用のデータはファイルとして（たとえば，SORT.INPというファイル名で）用意し，

A>SORT1 SORT.INP

のように実行時の引数として指定するようになっている。データファイルはリスト2のようなX-BASICプログラムで作成するのが手軽だろう。70～100行あたりの注釈で殺してある各行を適当に復活することで，作為的なデータを作ることもできる。

ソート結果は無条件にSORT.OUTというファイル名で出力される。あらかじめ動作が確認されているソートプログラムを使ってソートしたファイルを（たとえば,ORDERED.DATというファイル名で）

4) リスト1では162行の.end疑似命令で実行開始アドレスを指定しているから，リンク順序を変えても大丈夫。

リスト1 SORTTEST.S

```
  1: *
  2: *          ソートルーチンの動作試験用プログラム
  3: *
  4:            .include        doscall.mac
  5:            .include        iocscall.mac
  6:            .include        const.h
  7: *
  8:            .xref   sort
  9: *
 10: MAXDATA equ     8000                    *データの最大個数
 11: *
 12: FPACK   macro   callno                  *FLOATn.Xの
 13:         .dc.w   callno                  *  ファンクション呼び出し用
 14:         endm                            *  マクロ
 15: *
 16: __IUSING        equ     $fe18           *整数→文字列変換コール
 17:                                         *（桁数指定つき）
 18: *
 19:         .text
 20:         .even
 21: *
 22: ent:
 23:         lea.l   mysp,sp
 24:
 25:         bsr     readdata                *データを読み込む
 26:                                         *d7=配列の大きさ
 27:
 28:         IOCS    _ONTIME                 *計時開始
 29:         move.l  d0,d6                   *d6.l=仮の開始時刻
 30: tloop:  IOCS    _ONTIME                 *1/100秒単位の
 31:         cmp.l   d0,d6                   *  時間の変わり目を
 32:         beq     tloop                   *  検出する
 33:         move.l  d0,d6                   *d6=開始時刻
 34:
 35:         lea.l   sdata(pc),a0            *a0=ソートする配列
 36:
 37:         pea.l   0(a0,d7.l)              *配列末尾+1のアドレス
 38:         pea.l   (a0)                    *配列先頭のアドレス
 39:         bsr     sort                    *ソートする
 40:         addq.l  #8,sp                   *
 41:
 42:         IOCS    _ONTIME                 *時刻を得る
 43:
 44:         sub.l   d6,d0                   *d0=ソートに要した時間
 45:         bpl     tskip                   *
 46:         add.l   #24*60*60*100,d0
 47:                                         *負の場合は24時間分補正する
 48: tskip:  lea.l   temp(pc),a0             *ソートに要した時間を
 49:         moveq.l #7,d1                   *  10進7桁右詰めで
 50:         FPACK   __IUSING                *  文字列に変換する
 51:
 52:         bsr     conv                    *1/100秒単位から秒単位へ
 53:
 54:         pea.l   temp(pc)                *ソートに要した
 55:         DOS     _PRINT                  *  時間を表示する
 56:         pea.l   secmes(pc)              *
 57:         DOS     _PRINT                  *
 58:
 59:         bsr     writedata               *ソート結果を書き出す
 60:
 61:         DOS     _EXIT
 62:
 63: *
 64: *          データをファイルから読み込む
 65: *
 66: readdata:
 67:         move.w  #ROPEN,-(sp)
 68:         pea.l   1(a2)
 69:         DOS     _OPEN
 70:         addq.l  #6,sp
 71:         move.w  d0,d1
 72:         bmi     error
 73:
 74:         move.l  #MAXDATA*4,-(sp)
 75:         pea.l   sdata(pc)
 76:         move.w  d1,-(sp)
 77:         DOS     _READ
 78:         move.l  d0,d7
 79:         bmi     error
 80:
 81:         DOS     _CLOSE
 82:         lea.l   10(sp),sp
 83:
 84:         andi.l  #$ffff_fffc,d7  *d7=ソートする配列の総バイト数
 85:                                 *（ロングワード単位）
 86:         beq     error
 87:         rts
 88:
 89: *
 90: *          ソート結果をファイルに書き出す
 91: *
 92: writedata:
 93:         move.w  #ARCHIVE,-(sp)
 94:         pea.l   dfile(pc)
 95:         DOS     _CREATE
 96:         addq.l  #6,sp
 97:         move.w  d0,d1
 98:         bmi     error
 99:
100:         move.l  d7,-(sp)
101:         pea.l   sdata(pc)
102:         move.w  d1,-(sp)
103:         DOS     _WRITE
104:         cmp.l   d7,d0
105:         bne     error
106:
107:         DOS     _CLOSE
108:         lea.l   10(sp),sp
109:         rts
110:
111: *
112: *          エラー終了
113: *
114: error:
115:         pea.l   errmes(pc)
116:         DOS     _PRINT
117:
118:         move.l  #1,-(sp)
119:         DOS     _EXIT2
120:
121: *
122: *          1/100秒単位→秒単位変換
123: *
124: conv:
125:         lea.l   temp+7(pc),a1
126:         lea.l   temp+8(pc),a2
127:         moveq.l #$20,d1
128:         moveq.l #'0',d2
129:         clr.b   (a2)
130:         move.b  -(a1),-(a2)
131:         move.b  -(a1),d0
132:         cmp.b   d1,d0
133:         bne     skip1
134:         move.b  d2,d0
135: skip1:  move.b  d0,-(a2)
136:         move.b  #'.',-(a2)
137:         move.b  -(a2),d0
138:         cmp.b   d1,d0
139:         bne     skip2
140:         move.b  d2,(a2)
141: skip2:  rts
142: *
143:         .data
144:         .even
145: *
146: dfile:  .dc.b   'sort.out',0
147: errmes: .dc.b   'エラーが発生しました',CR,LF,0
148: secmes: .dc.b   ' sec.',CR,LF,0
149: *
150:         .bss
151:         .even
152: *
153: temp:   .ds.b   10                      *数値→文字列変換用
154: sdata:  .ds.l   MAXDATA                 *ソートデータ格納領域
155: *
156:         .stack
157:         .even
158: *
159: mystack:
160:         .ds.l   160000+256              *スタック領域
161: mysp:
162:         .end    ent
```

用意しておき，FC.Xを使って，

A>FC SORT.OUT ORDERED.DAT /B

というようにファイル内容を比較することで，ソートが正しく行えたかどうかが確認できる（バイナリファイルの比較を指定する/Bスイッチは必ずつけること）。

では，ここからが本題。

単純交換法

●アルゴリズム

ソートのもっとも原始的なアルゴリズムに，単純交換法がある。配列中の隣り合った要素を比べ，位置関係が狂っていたら交換するという操作を繰り返すことでソートしていく方法だ。

1）まず，配列末尾の2要素に注目し，大小関係が狂っていたらその位置を交換する。続いて，注目する位置を1つ手前にずらし，また，比較して，必要なら位置を交換する。以下，同様に配列先頭に達するまで比較・交換を繰り返す。図1に示すように，小さな値は順繰りに前に送られていく形になる。ここまでが1回分の“パス”。

2）この時点で先頭には配列中の最小値がある。先頭要素は確定したから，もう忘れてしまってよい。そこで，配列の2番目の要素以降に対して1)の操作を繰り返す。と，2番目に小さい値が配列2番目に移動する。

3）以下，同様に処理範囲を狭めつつ，同様の操作を繰り返す。残り要素が1個になったときにはソートが完了している。

以上のアルゴリズムの動きを追うと図2のようになる。小さな値がちょうど泡のように配列先頭のほうへ浮いていく様子から，このアルゴリズムはバブルソート（bubble sort）という名でも知られている（というより，こっちのほうがとおりがいい）。

●プログラム例

リスト3がバブルソートルーチンの一例だ。わけあって，パスの方向を上の説明とは逆にしてある。つまり，“配列の後ろのほうから比較・交換を繰り返して，先頭に最小値を移動させる”代わりに，配列の前のほうから比較・交換を繰り返して，末尾に最大値を移動している。

プログラムは単純な2重ループ構造をしている。24～33行のループがパス1回分の処理で，このループにより，最大値が処理範囲の末尾に移動する。それを23～37行のループで囲み，残り要素数が1個になるまで繰り返している。ここでは，サブルーチンに渡された配列の要素数が最初から1個しかない場合に備え，21行のbraによりループ先頭ではなくループの終了判定部分に飛び込んでいる点に注意したい。また，20行でa1（＝配列の末尾＋1を指している）を要素1個分減らしているのは，32行や37行のアドレス比較を楽にするための細工だ。

内側のループは，隣り合った要素の比較（24～26行），交換（29～30行）の2つの処理からなる。比較時には次のループに備えて，ポストインクリメント・アドレスレジスタ間接形式を使って，同時にポインタを進めている。そのため，29～30行の交換処理では，このずれの分をディスプレースメントで補正している。

ところで，ポインタを使ったメモリ操作時には，ポインタを増やすか減らすかによって，ポストインクリメント・アドレスレジスタ間接形式とプリデクリメント・アドレスレジスタ間接形式を使い分けるわけだが，68000では前者のほうが若干速い。パスの方向を逆にしたのは，うまくポストインクリメント・アドレスレジスタ間接形式が利用できるようにしたかったからにほかならない。

●改良案

バブルソートはその覚えやすい名前以上に，遅いアルゴリズムとして有名だ。少々改良したところでほかのソートアルゴリズムにはまるでかなわない。しかし，あっさり見捨てるのはかわいそうなので，できる限りの高速化をしてやろう。

リスト3では，処理範囲を狭めていき，残り要素数が1以下になった時点でソート完了と判断した。パス回数は常に配列の全要素数－1回だ。が，図2の例でもわかるように，実際にはその前にソートが

リスト2　GENDAT.BAS

```
 10 int fp,i,dat(7999)
 20 str s
 30 s = time$
 40 srand( atoi(right$(s,2)+mid$(s,4,2)+left$(s,2)) mod 32768 )
 50 for i=0 to 7999
 60    dat(i) = rand()            /*ランダム*/
 70 /* dat(i) = rand()/100        /*ランダム(重複率100倍)*/
 80 /* dat(i) = rand()/1000       /*ランダム(重複率1000倍)*/
 90 /* dat(i) = i                 /*ソートずみ*/
100 /* dat(i) = 7999-i            /*逆順ソートずみ*/
110 next
120 fp = fopen( "sort.inp", "c" )
130 fwrite( dat, 8000, fp )
140 fclose( fp )
```

```
51 45 76 23 38 16 83 60
51 45 76 23 38 16 60 83
51 45 76 23 38 16 60 83
51 45 76 23 16 38 60 83
51 45 76 16 23 38 60 83
51 45 16 76 23 38 60 83
51 16 45 76 23 38 60 83
16 51 45 76 23 38 60 83
```

図1　単純交換法における比較・交換

```
初期状態   51 45 76 23 38 16 83 60
           16|51 45 76 23 38 60 83
           16 23|51 45 76 38 60 83
           16 23 38|51 45 76 60 83
           16 23 38 45|51 60 76 83
           16 23 38 45 51|60 76 83
           16 23 38 45 51 60|76 83
ソート完了 16 23 38 45 51 60 76|83
```

図2　単純交換法（バブルソート）

完了する場合もある。ソートが完了したかどうかを早めに検出できれば，無駄な比較をしなくてもすみ，実行速度の向上が望めそうに思える。具体的には，交換が行われたかどうかを表すフラグを設け，ループ先頭でこのフラグを偽に初期化しておき，以後，交換が行われたらフラグを真にする。ループの最後でフラグが偽のままであれば，1回のパスのあいだに交換が行われなかったことがわかり，ソートずみと判断できる。

このとき，交換が行われたかどうかだけではなく，最後に交換が行われた位置も同時に記憶しておくと，さらに処理ステップの節約が図れる可能性がある。たとえば，

30 20 10 40 50 60|

のようなデータ列にバブルソートを適用する場合を考えよう（“|”で処理範囲の上限を表す）。最初のパスで，前から順に比較・交換していくと，まず30と20，さらに30と10の位置が交換されて，

20 10 30 40 50 60|
　　↑　↑

になる（矢印は最後に交換が行われた位置を表している）。ふつうなら，ここで処理範囲の上限を要素1個分狭めるわけだが，明らかに，“最後に交換が行われた位置より右”はすでにソートずみであり，処理範囲から除外してもよい。そこで，上限を一気に“最後に交換した2要素のあいだ”に動かす。

20 10|30 40 50 60
　　↑　↑

そうすれば，2度目のパスでは残りの2要素だけを相手にすればよく，素直に，

20 10 30 40 50|60

と上限を移動した場合よりも比較回数を減らすことができる。

リスト4に改良版を示す。いや，残念ながら改良ではなく，改悪になってしまった。なるたけの最適化を施したにも関わらず，平均的な性能はリスト3の版よりも多少落ちている。アルゴリズムを改良したことにより節約される比較時間よりも，新たにループ中に追加された命令の総実行時間のほうが長かったのだ。比較にかかる時間がもっと長い場合（たとえば，対象が整数データではなく文字列のとき）には，比較を1回省略するごとに節約できる時間も大きくなるから，改良の効果が出てくるはずなのだが。

せっかくだから，リスト4に施したマシン語流の最適化についても簡単に触れておく。まず，a0レジスタを交換が行われたかどうかを表すフラグと，最後の交換位置を指すポインタの両方に兼用してい

リスト3　SORT1.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           （バブルソート　あるいは　単純交換法）
 3: *           8000要素のソートに 231.33秒
 4: *
 5:             .xdef   sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l   1           *&array[0]（配列の先頭アドレス）
 9: ABOT:       .ds.l   1           *&array[N]（配列の最終アドレス+1）
10: *
11:             .text
12:             .even
13: *
14: sort:
15:             link    a6,#0
16:             movem.l d0-d1/a0-a2,-(sp)
17:
18:             movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                     *a1=配列末尾+1のアドレス
20:             subq.l  #4,a1           *a1=最終要素のアドレス
21:             bra     next1
22:
23: loop1:      movea.l a0,a2           *a2=注目している位置
24: loop2:      move.l  (a2)+,d0        *隣り合う
25:             move.l  (a2),d1         *  2つの要素を
26:             cmp.l   d1,d0           *  比較して
27:             ble     next2           *  順序の乱れを調べる
28:
29:             move.l  d1,-4(a2)       *順序が乱れていたら
30:             move.l  d0,(a2)         *  位置を交換する
31:
32: next2:      cmpa.l  a1,a2           *上限に達するまで
33:             bcs     loop2           *  繰り返す
34:
35:             subq.l  #4,a1           *上限を狭める
36: next1:      cmpa.l  a1,a0           *上限が配列先頭に達するまで
37:             bcs     loop1           *  繰り返す
38:
39: done:       movem.l (sp)+,d0-d1/a0-a2
40:             unlk    a6
41:             rts
42:
43:             .end
```

リスト4　SORT2.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           （バブルソート：改良しようとして改悪した版）
 3: *           8000要素のソートに 237.74秒
 4: *
 5:             .xdef   sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l   1           *&array[0]（配列の先頭アドレス）
 9: ABOT:       .ds.l   1           *&array[N]（配列の最終アドレス+1）
10: *
11:             .text
12:             .even
13: *
14: sort:
15:             link    a6,#0
16:             movem.l d0-d2/a0-a3,-(sp)
17:
18:             movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                     *a1=配列末尾+1のアドレス
20:             subq.l  #4,a1           *a1=最終要素のアドレス
21:
22:             cmpa.l  a1,a0           *要素数が1以下なら
23:             bcc     done            *  ソート済み
24:
25:             moveq.l #0,d2           *d2=0
26:             addq.l  #4,a1           *すぐ下の行のための逆補正
27: loop1:      lea.l   -4(a1),a3       *a3=処理の上限
28:             movea.l a0,a2           *a2=注目している位置
29:             movea.l d2,a1           *a1=0
30:
31: loop2:      move.l  (a2)+,d0        *隣り合う
32:             move.l  (a2),d1         *  2つの要素を
33:             cmp.l   d1,d0           *  比較して
34:             ble     next2           *  順序の乱れを調べる
35:
36:             move.l  d1,-4(a2)       *順序が乱れていたら
37:             move.l  d0,(a2)         *  位置を交換する
38:             movea.l a2,a1           *a1=次のループの上限+4
39:
40: next2:      cmpa.l  a3,a2           *上限に達するまで
41:             bcs     loop2           *  繰り返す
42:
43: next1:      cmpa.l  d2,a1           *交換が行われなくなるまで
44:             bne     loop1           *  繰り返す
45:
46: done:       movem.l (sp)+,d0-d2/a0-a3
47:             unlk    a6
48:             rts
49:
50:             .end
```

る。a0が0のときは“交換がまだ行われていない”ことを表す。交換が行われたら，その位置をa0に入れ，そのことによってa0は非0となり，“交換が行われたこと”を表すようになる。この周辺では，d2レジスタにあらかじめ0を入れておき，

```
suba.l     a1,a1
```

の代わりに，

```
movea.l    d2,a1
```

を，

```
cmpa.l     #0,a1
```

の代わりに，

```
cmpa.l     d2,a1
```

を使うといった，小さな細工もしてある。

また，38行に注目してもらいたい。よく見てもらうと，a1に格納されるアドレスは要素1個のバイト数だけ（ここでは4）予定よりずれている。この分の補正は，外側のループの先頭27行の，

```
lea.l      -4(a1),a3
```

で行っている。それにともない，ループに入る直前に，

```
addq.l     #4,a1
```

を入れて逆補正していたりもする。わざわざこんなことをしているのは次の理由による。

アルゴリズムどおりの動作をさせるためには，27行か38行のどちらかを，

```
lea.l      -4(aX),aX
```

に，逆側を，

```
movea.l    aX,aX
```

にする必要がある。ここで，前者は後者よりも多少遅い。となれば，“頻繁に実行されるほう”により速い命令を使うべきだ。27行と38行の実行回数を考えると，27行はパス1回ごとに1度実行され，38行はパスの中で交換が発生するたびに1度実行される。多くの場合，

パス回数＜＜交換回数

だから，交換時の処理時間を短縮するほうが速度的に有利，となる。つじつま合わせに追加された26行は，ループの外側でただ1回実行されるだけだから，大勢には影響しない。

●シェーカーソート

改悪になってしまったのが悔しいので，少しムキになって，さらにアルゴリズムの改良に取り組む。

60 10 20 30 40 50

というデータ列をリスト4の改悪版バブルソートルーチンでソートすることを考えよう。最初のパスで5回の交換が起こり，最大値60は配列末尾に移動して，1パスでソートが完了する。正確には，ソートが完了したかどうか知るためにはもう一度走査する必要があるが，それでも2パスだ。では，

20 30 40 50 60 10

というデータ列ではどうだろう。パス1回ごとに交換がただ1度行われ，末尾に置かれた最小値の10は順々に前に送られていく。都合5パスだ。交換回数は上の例と変わらないが，比較回数はずっと多くなってしまう。

では，パス方向を逆にして（配列の後ろのほうから比較・交換を繰り返して，先頭に最小値を移動させる），同じデータ列をソートしてみる。今度は状況は逆転し，

20 30 40 50 60 10

は2パスでソートできるが，

60 10 20 30 40 50

をソートするのに5回のパスが必要になる。バブルソートにはアルゴリズム上の偏りがあり，パスの方向によって，得意なデータと不得手なデータがはっきり分かれるわけだ。

リスト5　SORT3.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (シェーカーソート)
 3: *           8000要素のソートに 179.26秒
 4: *
 5:          .xdef    sort
 6: *
 7:          .offset 8
 8: ATOP:    .ds.l    1        *&array[0](配列の先頭アドレス)
 9: ABOT:    .ds.l    1        *&array[N](配列の最終アドレス+1)
10: *
11:          .text
12:          .even
13: *
14: sort:
15:          link     a6,#0
16:          movem.l d0-d2/a0-a3,-(sp)
17:
18:          movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                  *a1=配列末尾+1のアドレス
20:          subq.l   #4,a1          *a1=最終要素のアドレス
21:
22:          cmpa.l   a1,a0          *要素数が1以下なら
23:          bcc      done           *  ソート済み
24:
25:          moveq.l #0,d2           *d2=0
26:
27: loop1:   movea.l a0,a3           *a3=処理の下限
28:          movea.l a1,a2           *a2=注目している位置
29:          movea.l d2,a0           *a0=0
30:
31: loop2:   move.l   (a2),d0        *隣り合う
32:          move.l   -(a2),d1       *  2つの要素を
33:          cmp.l    d1,d0          *  比較して
34:          bge      next2          *  順序の乱れを調べる
35:
36:          move.l   d1,4(a2)       *順序が乱れていたら
37:          move.l   d0,(a2)        *  位置を交換する
38:          movea.l a2,a0           *a0=次のループの下限
39:                                  *(本当はもう1要素分狭められる)
40: next2:   cmpa.l   a2,a3          *下限に達するまで
41:          bcs      loop2          *  繰り返す
42:
43:          cmpa.l   d2,a0          *交換が行われなくなるまで
44:          beq      done           *  繰り返す
45:
46:          movea.l a1,a3           *a3=処理の上限
47:          movea.l a0,a2           *a2=注目している位置
48:          movea.l d2,a1           *a1=0
49:
50: loop3:   move.l   (a2)+,d0       *隣り合う
51:          move.l   (a2),d1        *  2つの要素を
52:          cmp.l    d1,d0          *  比較して
53:          ble      next3          *  順序の乱れを調べる
54:
55:          move.l   d1,-4(a2)      *順序が乱れていたら
56:          move.l   d0,(a2)        *  位置を交換する
57:          movea.l a2,a1           *a1=次のループの上限
58:                                  *(本当はもう1要素分狭められる)
59: next3:   cmpa.l   a3,a2          *上限に達するまで
60:          bcs      loop3          *  繰り返す
61:
62: next1:   cmpa.l   d2,a1          *交換が行われなくなるまで
63:          bne      loop1          *  繰り返す
64:
65: done:    movem.l (sp)+,d0-d2/a0-a3
66:          unlk     a6
67:          rts
68:
69:          .end
```

そこで，1回のパスごとにパス方向を変えて，偏りをならせば，平均的な性能も向上するのではないかという発想が生まれてくる。このバブルソートの変形はシェーカーソート（shaker sort）というこれまた冗談のような名前で知られている。リスト4をベースに作り直したプログラムをリスト5に示そう。今度はちゃんと改良になった。

プログラムは，パスの方向を変えた2つのループを直列に並べてあるというだけの代物で，見るべきところはなにもない。すでにバブルソートに対しては十分な礼儀をつくしたことでもあるし，そろそろ飽きてもきたので，下限や上限を更新する処理は若干手抜きスタイルになっていたりもする。リスト4同様，38行や57行でアドレスレジスタに入れているアドレスは望む値より4バイトずれているが，この補正をどこでもやっていない。これにより，常に下限や上限が要素1個分ずれており，パス1回ごとに余計な比較が1度ずつ行われる。気になる人は改良してもらいたい。

単純選択法

●アルゴリズム

単純選択法も，バブルソート同様，パス1回ごとに最小値を1つ先頭に移動することで，配列の端から順にソートしていく方法だ。ただ，最小値を移動するとき，ほかの要素をずらす代わりに，単純に，先頭要素と最小値の位置を交換する。

1)　最初は配列全体を検索範囲とし，検索範囲中から最小値を探す。見つけた最小値は検索範囲の先頭要素と位置を交換する。ここまでが1回分のパス。

2)　先頭要素は確定した。検索範囲の下限をずらし，第2要素以降からふたたび最小値を探して，検索範囲の先頭と交換する。

3)　以下，同様に検索範囲を狭めつつ，最小値を探しては先頭と交換する。残り要素が1個になったときにはソートが完了している。

図3に動作の様子を示す。

●プログラム例

最小値を探す処理は，検索範囲の端の要素を仮の最小値としてレジスタに保管しておき，以降の要素と順次比較して，“仮の最小値”より小さい値を見つけたらレジスタに拾い上げる（仮の最小値とする），という操作を繰り返すことで行える。

リスト6が単純選択法のプログラム例だ。いくつか作ったなかから比較的綺麗な版を選んでみた（その割にはループ構造がひねている）。

外側のループは25行から始まる。残り要素数が1以下かどうか（ソートが完了したかどうか）を調べてから，28～29行で仮の最小値をd0に，その位置をa3に入れておき，30～33行の内側のループで順次比較を繰り返す。検索方向は検索範囲の後ろから前方向に行っている（逆にした版はどうも収まりが悪かった）。ひととおり比較し終わった時点でd0に真の最小値，a3にその格納位置が残っているから，23～24行で，検索範囲の先頭要素と交換する。

単純挿入法

単純挿入法は，配列を“いちおう順序が整った部分”と“そうでない部分”に分けて，未整理の部分から1つ要素を取り出しては整理ずみの部分の適当な位置に挿入することでソートしていく。ある意味で，バブルソートや単純選択法とは逆の発想に基づくアルゴリズムといえる。

●アルゴリズム

1)　最初は配列の先頭1要素だけが整理ずみ，残りが未整理と考える（要素が1個しかない配列は常にソートずみである）。

2)　未整理部分の先頭，つまりは，配列の2番目の要素を抜き出し，これを整理ずみの部分の適当な位置に挿入する。ここまでが1パス。

3)　2)の時点で整理ずみの部分は要素1個分伸び，未整理部分は1個分削られた。ふたたび，未整理部

図3　単純選択法

初期状態	51 45 76 23 38 16 83 60
	16|45 76 23 38 51 83 60
	16 23|76 45 38 51 83 60
	16 23 38|45 76 51 83 60
	16 23 38 45|76 51 83 60
	16 23 38 45 51|76 83 60
	16 23 38 45 51 60|83 76
ソート完了	16 23 38 45 51 60 76|83

リスト6　SORT4.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (単純選択法)
 3: *           8000要素のソートに 131.08秒
 4: *
 5:             .xdef   sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l   1           *&array[0](配列の先頭アドレス)
 9: ABOT:       .ds.l   1           *&array[N](配列の最終アドレス+1)
10: *
11:             .text
12:             .even
13: *
14: sort:
15:             link    a6,#0
16:             movem.l d0/a0-a3,-(sp)
17:
18:             movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                     *a1=配列末尾+1のアドレス
20:             subq.l  #4,a1           *a1=最終要素のアドレス
21:             bra     loop1
22:
23: swap:       move.l  (a0),(a3)       *みつけた最小値と
24:             move.l  d0,(a0)+        *  先頭要素を交換する
25: loop1:      cmpa.l  a1,a0           *配列の最後に達した?
26:             bcc     done            *  そうならソート完了
27:             movea.l a1,a2           *a2=検索位置
28: select:     move.l  (a2),d0         *d0=仮の最小値
29:             movea.l a2,a3           *a3=d0が格納されていた位置
30: loop2:      cmpa.l  a2,a0           *配列の先頭に達した?
31:             bcc     swap            *  d0に入っているのが最小値
32:             cmp.l   -(a2),d0        *仮の最小値より小さい値を
33:             ble     loop2           *  探し続ける
34:             bra     select          *みつかったら仮の最小値にする
35:
36: done:       movem.l (sp)+,d0/a0-a3
37:             unlk    a6
38:             rts
39:
40:             .end
```

分の先頭＝配列の3番目の要素を抜き出し，整理ずみの部分の適当な位置に挿入する。
4) パス1回ごとに未整理の要素が1個ずつ減っていく。それが0になったらソートは完了する。

以上の様子を図4に示す。

●プログラム例

挿入位置を探す処理は，単純に，整理ずみの範囲の端から順に比較を繰り返すことで行える。このとき，整理ずみの範囲の末尾から先頭方向に比較していくことにすれば，挿入しようとするデータよりも大きな値をどんどん後ろにずらしていくことで，挿入位置の検索とデータの移動を並行して進めることができる（図5）。

では，リスト7にプログラムの実例を示そう。例によってプリデクリメント・アドレスレジスタ間接形式を嫌う方針により，パスの方向を逆にして，整理ずみの部分は配列先頭ではなく，配列の末尾におくようにしてある。a2が整理部分と未整理部分の境界を示し，a2以降が整理ずみだ。挿入位置はそのa2から後ろに向かって探していく形になる。ほかの部分に若干しわよせがいっているが，内側のループ中もっとも頻繁に実行される27行にポストインクリメント・アドレスレジスタ間接形式を適用するためならなんでもありだ。

引数の取り込み後，20行でまず整理ずみ部分と未整理部分の境界a2を初期化する。最初は配列の末尾の要素ただ1個が整理ずみの領域に入り，残りが未整理となる。

22行から外側のループが始まる。ここでは，挿入位置を探すポインタとして使うa3を初期化して（初期値は整理ずみ部分の先頭＝a2）から，23行で境界をずらすと同時に，新規に挿入するデータをd0に取り込む。その後，24行で境界が配列の先頭を越えたかどうかを調べ，越えたとなればソート完了と判断してループを抜ける。配列先頭とのアドレス比較の前にデータを取り込んでしまっているのは反則[5)]なのだが，大目に見てもらいたい。

以下，27行からが内側のループ。ここではa3を順次進めつつ，挿入すべきデータとの比較を繰り返し，挿入位置を探している。挿入位置が見つかるまでは31行で挿入場所を作るためにデータをずらし，挿入位置が見つかったら38行に飛び，適切な位置にデータを収める。どちらの場合も，27行の比較の時点でポインタをポストインクリメントしている関係で，a3は挿入位置を派手に通り過ぎている点に注意してもらいたい。

図4　単純挿入法

初期状態	51｜45 76 23 38 16 83 60
	45 51｜76 23 38 16 83 60
	45 51 76｜23 38 16 83 60
	23 45 51 76｜38 16 83 60
	23 38 45 51 76｜16 83 60
	16 23 38 45 51 76｜83 60
	16 23 38 45 51 76 83｜60
ソート完了	16 23 38 45 51 60 76 83｜

ところで，挿入データがどの整理ずみデータよりも大きかったときには，挿入位置が見つからないまま配列の端に達することになる。32行のアドレス比較がこのためのチェックであり，このチェックに引っ掛かった場合は速やかにループを抜けて35行で配列の最後尾にデータを格納する。

●改良案

リスト7では内側のループの脱出条件に次の2通りがある。

1) 挿入位置が見つかった（挿入する値以上のデータが見つかった）。

図5　単純挿入法における挿入位置の検索

この状態から60の挿入位置を探す	16 23 38 45 51 76 83 60
↓	
60<83だから挿入位置はもっと前	16 23 38 45 51 76 83 □ 60
↓	
60<76だから挿入位置はもっと前	16 23 38 45 51 76 □ 83 60
↓	
60>51だから挿入位置はもっと後	16 23 38 45 51 □ 76 83 60
↓	
ということはここ	16 23 38 45 51 60 76 83

表1　ソート時間（単位：秒）

		n＝500	n＝1000	n＝2000	n＝4000	n＝8000
バブルソート	A	0.90	3.60	14.46	57.76	231.33
	B	0.89	3.59	14.45	57.84	231.05
	C	0.89	3.60	14.39	57.34	229.81
	D	0.73	2.94	11.77	47.10	188.41
	E	1.07	4.28	17.15	68.64	274.57
	F	0.73	2.94	11.77	47.10	188.41
バブルソート（改良したつもりだった版）	A	0.92	3.70	14.87	59.45	237.74
	B	0.92	3.65	14.84	59.47	237.81
	C	0.91	3.67	14.75	58.84	236.18
	D	0.00	0.00	0.01	0.02	0.04
	E	1.12	4.50	18.00	72.03	288.12
	F	0.00	0.00	0.01	0.02	0.04
シェーカーソート	A	0.70	2.78	11.20	44.64	179.26
	B	0.68	2.77	11.24	44.88	178.18
	C	0.69	2.81	11.05	43.56	175.44
	D	0.00	0.00	0.01	0.02	0.04
	E	1.14	4.56	18.24	72.92	291.60
	F	0.00	0.00	0.01	0.02	0.04
単純選択法	A	0.52	2.07	8.23	32.82	131.08
	B	0.52	2.06	8.22	32.80	130.98
	C	0.51	2.06	8.21	32.76	130.91
	D	0.83	3.34	13.35	53.41	213.59
	E	0.67	2.69	10.77	43.06	172.21
	F	0.51	2.05	8.18	32.72	130.84
単純挿入法	A	0.38	1.52	6.15	24.37	98.00
	B	0.37	1.50	6.13	24.54	97.37
	C	0.37	1.51	5.98	23.40	94.57
	D	0.00	0.00	0.01	0.03	0.07
	E	0.75	3.03	12.13	48.54	194.16
	F	0.00	0.00	0.01	0.03	0.07
単純挿入法（番人あり）	A	0.34	1.36	5.52	21.88	88.00
	B	0.33	1.34	5.50	22.04	87.43
	C	0.33	1.35	5.37	21.01	84.91
	D	0.00	0.00	0.01	0.03	0.06
	E	0.68	2.72	10.90	43.60	174.36
	F	0.00	0.00	0.01	0.03	0.06

A：ランダム
B：ランダム（同一データが重複している確率がAの100倍）
C：ランダム（同1000倍）
D：ソートずみ
E：逆順にソートずみ
F：全要素が0

5) 24〜25行のチェックに引っ掛かってループを抜ける直前には，すでに23行の時点でa2は配列の領域外を指している。読むだけで書き換えるわけではないとはいえ，引数として渡された配列の外にアクセスするのは不作法だ。

2) 挿入位置が見つからないまま配列の端に達した。

ループ1回につき，条件判断が2つあるのは実行速度の点でも，プログラムの簡潔さという点でもあまりよいことではない。このような場合には，“番人”を置いて，ループの脱出条件をひとつにすることができないかどうかを考えてみるべきだ。

番人（sentinel：番兵，見張り，標識などとも訳される）とは，アルゴリズムの導入上，仮に追加する終端データのことをいう。いま，ソートする配列のすぐ後ろに，配列中の最大値以上の値（それが番人）を置いてみる。たったそれだけのことで，上の2)のチェックは不要となる。挿入位置の検索が配列の端に達したときには，番人との比較が行われることになり，必ず1)のチェックに引っ掛かるからだ。

リスト7の21行のあたりに番人をセットする1行を付け加え，32～36行をばっさり削って，

```
        bra         loop2
```

に置き換えたうえで，ループ構造やレジスタの使い方を多少最適化したらリスト8のようになった。

20行で番人をセットしている。いま扱うデータは符号つき32ビット数だから，その最大値である7FFFFFFF$_H$を番人に使う。ここで，サブルーチンに渡された配列の直後には番人を収めるだけの空きがあることを仮定している。このサブルーチンを呼び出すときには，呼び出し側がその分のメモリを確保しておかなければならない。その意味で，番人を利用するアルゴリズムはメインルーチンに余計な負担をかけるといえる。安全を期すなら，リスト1のテストプログラムも，154行を，

```
        sdata:    .ds.l    MAXDATA+1
```

に修正してアセンブルし直したほうがよいだろう（直後は未使用のスタック領域だから，修正しなくても実害はないが）。

＊

とりあえず，今回作った各プログラムに対して，いろいろな種類・大きさのデータを与えてみたときの実行時間を表1に示しておく。表中A～Fはテストデータの傾向で，A～Eがリスト2の70～100行にそれぞれ対応する。

どのアルゴリズムも，要素数（nとおく）を倍にすると，実行時間が4倍になるのがわかると思う。実行時間はn^2に比例し，nが増えると放物線を描いて急激に上昇していく。後編では，この上昇カーブがずっと緩やかなソートアルゴリズム，シェルソート，ヒープソート，クイックソートが登場する。今月紹介した単純なアルゴリズムとの性能の差は歴然としていて，クイックソートに至っては，8000個のデータのソートに1秒とかからない。では。

リスト7 SORT5.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (単純挿入法)
 3: *           8000要素のソートに 98.00秒
 4: *
 5:         .xdef   sort
 6: *
 7:         .offset 8
 8: ATOP:   .ds.l   1           *&array[0](配列の先頭アドレス)
 9: ABOT:   .ds.l   1           *&array[N](配列の最終アドレス+1)
10: *
11:         .text
12:         .even
13: *
14: sort:
15:         link    a6,#0
16:         movem.l d0-d1/a0-a3,-(sp)
17:
18:         movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                 *a1=配列末尾+1のアドレス
20:         lea.l   -4(a1),a2       *a2=整理済み部分と未整理部分の境界
21:
22: loop1:  movea.l a2,a3           *a3=挿入位置を指すポインタ
23:         move.l  -(a2),d0        *d0=これから挿入するデータ
24:         cmpa.l  a0,a2           *未整理部分はもうない?
25:         bcs     done            * そうなら終了
26:
27: loop2:  move.l  (a3)+,d1        *ここは
28:         cmp.l   d1,d0           * 挿入位置?
29:         ble     found           * そうならループを抜ける
30:
31:         move.l  d1,-8(a3)       *のちの挿入時に備えて要素をずらす
32:         cmpa.l  a1,a3           *配列の端に達するまで
33:         bcs     loop2           * 繰り返す
34:
35: insert: move.l  d0,-(a3)        *端につけ足す
36:         bra     loop1
37:
38: found:  move.l  d0,-8(a3)       *挿入する
39:         bra     loop1
40:
41: done:   movem.l (sp)+,d0-d1/a0-a3
42:         unlk    a6
43:         rts
44:
45:         .end
```

リスト8 SORT6.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (単純挿入法:番人を置く版)
 3: *           8000要素のソートに 88.00秒
 4: *
 5:         .xdef   sort
 6: *
 7:         .offset 8
 8: ATOP:   .ds.l   1           *&array[0](配列の先頭アドレス)
 9: ABOT:   .ds.l   1           *&array[N](配列の最終アドレス+1)
10: *
11:         .text
12:         .even
13: *
14: sort:
15:         link    a6,#0
16:         movem.l d0-d1/a0-a2,-(sp)
17:
18:         movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                 *a1=配列末尾+1のアドレス
20:         move.l  #$7fffffff,(a1) *番人
21:         subq.l  #4,a1           *a1=整理済み部分と未整理部分の境界
22:         bra     next1
23: loop1:
24: loop2:  move.l  (a2)+,d1        *ここは
25:         cmp.l   d1,d0           * 挿入位置?
26:         ble     found           * そうならループを抜ける
27:
28:         move.l  d1,-8(a2)       *のちの挿入時に備えて
29:                                 * 要素をずらしていく
30:         bra     loop2           *挿入位置が見つかるまで
31:                                 * 繰り返す
32: found:  move.l  d0,-8(a2)       *挿入する
33:
34: next1:  movea.l a1,a2           *a2=挿入位置を指すポインタ
35:         move.l  -(a1),d0        *d0=これから挿入するデータ
36:         cmpa.l  a0,a1           *未整理部分がなくなるまで
37:         bcc     loop1           * 繰り返す
38:
39: done:   movem.l (sp)+,d0-d1/a0-a2
40:         unlk    a6
41:         rts
42:
43:         .end
```

X68000用 Misty Blueより オープニングテーマ曲

Tachikawa Masayuki 立川 正之

X1/turbo用 スプーンおばさんより リンゴの森の子猫たち

Kato Takashi 加藤 隆

正月気分もすっかり抜けて，またあわただしい日常が戻ってきましたが，皆さんいかがお過ごしでしょうか。さて，今月はエニックスのゲームミュージック1本と，アニメ「スプーンおばさん」のエンディングテーマをお届けします。こたつでみかんでも食べながらのんびりと打ち込んで，楽しんでください。

Yの悲劇(?)，主演：立川正之

お待たせしました。久しぶりに登場の立川君です。良い子のためのFM音源講座やYコマンドで（?）もうお馴染みですよね。今回の作品では彼らしいYコマンドの使い方がみられませんが，曲のデキは折り紙付きです。

さて，その曲はエニックスのアドベンチャーゲーム，Misty Blueのオープニングテーマです。このゲームはPC-8801用ですので，本紙の読者には馴染みが薄いタイトルかもしれません。曲の雰囲気は，ひと昔前に流行っていた，“SHOW ME”みたいな感じです。その曲をタイトルとしていた“男女7人秋物語”が恋愛ドラマだったことも考えると，Misty Blueはここいらへんを意識しているのかもしれませんね。

この作品ではパンポットが命ともいえそうで，ステレオ感がとてもうまく表現されています。ヘッドホンか，左右がきっちりと離れているスピーカーの定位置で聴きましょう。なお，OPMA用に調整されてあるようですので，ボスコニアンのサンプリングデータをお持ちの人はそちらで聴いてください。

ワンポイント・テクニックとして，ファンクションキーの19，20にテンポを指定するコマンドを入れているようですね。試しに演奏中にシフトキーとF9キーを押してみてください。早送りになったでしょう。おそらく，制作中にひんぱんに使うのでしょうが，確かに便利です。こういったファンクションキーの有効利用は大いに結構ですが，F1～F10は皆さんそれぞれがある程度設定し直して使っていることでしょう。もし，プログラムに残すときはF11～F20とか普段あまり使わないようなところに入れておくほうがよいようですね。

Misty Blue

今日はおばさんのバースディ

X1用にはNHKのアニメ「スプーンおばさん」より，「リンゴの森の子猫たち」をお送りしましょう。これはこの番組のエンディングテーマだったものです。8年前というかなり昔のモノですので，知らない人も多いかもしれませんね。そのころ中学生だった私がこうして原稿を書いているのですから，時がたつのは早いものです。

プログラムはMusic BASIC用です。短いし，使い回しがきく行もありそうですから，入力もかなり楽でしょう。曲を聴いていると，おばさんが等身大のスプーンと躍っているシーンが思い出されます。明るくて，とっても楽しい曲ですよ。小節によってはパートがかなり空いているようなので，いろいろと遊んでみるのもいいかもしれませんね。

さて，新年早々にやっためぞん一刻の反響で「X1でもなんかやれ～」という，なかなか無責任なリクエストが届いています。全体的に見るとX1のほうがジャンルも豊富だし，投稿者の層も多彩です。「やればできる」ということを知っているユーザーが多い事もあるのでしょうが，「やらないとアブナイ」という意識もあるのでしょう。やはり自分で作って投稿していただくと，X1用のプログラムも増えて嬉しかったりするんです（なんといっても私もX1ユーザー）。加藤君もX1の投稿の少なさを嘆いて投稿してくれました。皆さんもがんばってください。

それでは来月もまたこの雑誌のこのページでお会いしましょう。　(S.K.)

リスト1　Misty Blue

```
 10 /*                                                       sa."blue
 20 /*                         とにもかくにも
 30 /*
 40 /*                    「M i s t y   B l u e」
 50 /*
 60 /*                    なのであります。はにゃ～。
 70 /*
 80 /*       P r o g r a m e d   b y   立川正之  in 1990/12/9
 90 /*
100 dim char v(4,10)={
110 /*  AF    OM    WF    SY    SP   PMD   AMD   PMS   AMS   PAN
120     58,   15,    0,    0,    0,    0,    0,    0,    0,    3,    0,
130 /*  AR    DR    SR    RR    SL    OL    KS    ML   DT1   DT2   AME
140     31,    0,    0,    0,    0,   33,    0,    3,    0,    0,    0,
150     31,    0,    0,    0,    0,   55,    0,   10,    0,    1,    0,
160     31,    0,    0,    0,    0,   33,    0,    1,    0,    0,    0,
170     31,   16,    0,    4,   10,    3,    0,    3,    0,    0,    0}
180 m_vset(71,v)
190 /*
200 v={
210 /*  AF    OM    WF    SY    SP   PMD   AMD   PMS   AMS   PAN
220     60,   15,    0,    0,    0,    0,    0,    0,    0,    3,    0,
230 /*  AR    DR    SR    RR    SL    OL    KS    ML   DT1   DT2   AME
240     31,   10,    0,   12,   15,   30,    0,    1,    7,    0,    0,
250     31,   10,    0,    2,   15,    0,    0,    2,    7,    0,    0,
260     31,   20,    0,   12,   15,    0,    0,    2,    3,    0,    0,
270     31,   10,    0,    2,   15,    0,    0,    4,    3,    0,    0}
280 m_vset(72,v)
290 /*
300 v={
310 /*  AF    OM    WF    SY    SP   PMD   AMD   PMS   AMS   PAN
320      8,   15,    0,    0,    0,    0,    0,    0,    0,    3,    0,
330 /*  AR    DR    SR    RR    SL    OL    KS    ML   DT1   DT2   AME
340      5,    0,    0,    0,    0,   15,    0,    0,    0,    0,    0,
350      6,    0,    0,    0,    0,   25,    0,    2,    0,    0,    0,
360      7,    0,    0,    0,    0,   25,    0,    2,    0,    0,    0,
370     28,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0}
380 m_vset(73,v)
390 /*
400 v={
410 /*  AF    OM    WF    SY    SP   PMD   AMD   PMS   AMS   PAN
420     60,   15,    0,    0,    0,    0,    0,    0,    0,    3,    0,
430 /*  AR    DR    SR    RR    SL    OL    KS    ML   DT1   DT2   AME
440     31,    0,    0,   14,    0,   24,    0,    4,    3,    0,    0,
450     15,   15,    0,   14,    2,    2,    0,    2,    3,    0,    0,
460     31,    0,    0,   14,    0,   29,    0,    8,    7,    0,    0,
470     15,   15,    0,   14,    2,   10,    0,    4,    7,    0,    0}
480 m_vset(74,v)
```

日本音楽著作権協会(出)許諾第9072264-001号

```
490 /*
500 v={
510 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
520     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
530 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
540     31,   0,   0,   0,   0,  24,   0,   8,   3,   0,   0,
550     31,  16,   0,   7,   5,   0,   0,   4,   3,   0,   0,
560     31,   0,   0,   0,   0,  29,   0,   4,   7,   0,   0,
570     31,  16,   0,   7,   5,   8,   0,   2,   7,   0,   0}
580 m_vset(75,v)
590 /*
600 v={
610 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
620     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
630 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
640     26,  10,   0,   5,   2,  30,   1,   2,   0,   0,   0,
650     26,  10,   0,   8,   2,  15,   1,  10,   0,   0,   0,
660     26,  10,   0,   6,   2,  50,   1,   2,   0,   0,   0,
670     16,   0,   0,   6,   2,   2,   1,   2,   0,   0,   0}
680 m_vset(76,v)
690 /*
700 v={
710 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
720     58,  15,   3,   1, 237,   0,  88,   0,   1,   3,   0,
730 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
740     30,   4,   0,   3,   1,  37,   2,   1,   7,   0,   0,
750     25,   9,   1,   0,   1,  47,   2,  12,   0,   0,   0,
760     30,   4,   3,   4,   1,  37,   1,   3,   3,   0,   0,
770     20,   7,   4,   3,   5,   0,   2,   1,   0,   0,   0}
780 m_vset(77,v)
790 /*
800 v={
810 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
820      0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
830 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
840      0,   0,   0,  15,   0, 127,   0,   0,   0,   0,   0,
850      0,   0,   0,  15,   0, 127,   0,   0,   0,   0,   0,
860      0,   0,   0,  15,   0, 127,   0,   0,   0,   0,   0,
870      0,   0,   0,  15,   0, 127,   0,   0,   0,   0,   0}
880 m_vset(78,v)
890 /*
900 v={
910 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
920     36,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
930 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
940     15,  10,   0,   4,   3,  15,   1,   8,   3,   0,   0,
950     15,  15,   0,   4,   3,   5,   1,   8,   3,   0,   0,
960     15,  10,   0,   4,   3,  10,   1,   6,   7,   0,   0,
970     25,  15,   0,   4,   3,   5,   1,   6,   7,   0,   0}
980 m_vset(79,v)
990 /*
1000 v={
1010 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1020     59,  15,   3,   1, 237,   0,  88,   0,   1,   3,   0,
1030 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1040     31,  21,  19,   6,   2,   0,   0,  15,   2,   0,   0,
1050     31,  21,  12,   6,   2,  35,   0,   8,   2,   0,   0,
1060     31,  21,  13,   6,   3,  32,   0,   7,   3,   0,   0,
1070     31,  19,  16,   9,   2,   0,   0,   2,   3,   0,   0}
1080 m_vset(80,v)
1090 /*
1100 v={
1110 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1120     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1130 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1140     31,   0,   0,   0,   0,  24,   0,   8,   3,   0,   0,
1150     31,  16,   6,   8,   5,   0,   0,   4,   3,   0,   0,
1160     31,   0,   0,   0,   0,  29,   0,   4,   7,   0,   0,
1170     31,  16,   6,   8,   5,   4,   0,   2,   7,   0,   0}
1180 m_vset(81,v)
1190 /*
1200 v={
1210 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1220     28,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1230 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1240     31,  14,   0,  15,  15,  33,   3,  11,   6,   0,   0,
1250     30,   9,   7,   6,   1,  22,   3,   1,   3,   0,   0,
1260     31,  16,   5,   0,  10,  22,   1,   7,   3,   0,   0,
1270     30,   7,   4,   7,   1,   2,   3,   1,   0,   0,   0}
1280 m_vset(82,v)
1290 /*
1300 v={
1310 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1320     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1330 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1340     12,   7,   2,   8,   3,  23,   2,   1,   3,   0,   0,
1350     16,   7,   0,   8,   3,  32,   1,   3,   5,   0,   0,
1360     20,   7,   0,   8,   3,  42,   0,   1,   3,   0,   0,
1370     18,   7,   0,   8,   4,   0,   1,   1,   0,   0,   0}
1380 m_vset(83,v)
1390 /*
1400 v={
1410 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1420      7,  15,   3,   1, 237,   0,  88,   0,   1,   3,   0,
1430 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1440      8,   8,   8,   8,   8,   8,   3,   8,   0,   8,   8,
1450      8,   8,   8,   8,   8,   8,   3,   8,   3,   8,   8,
1460      8,   8,   8,   8,   8,   8,   3,   8,   4,   8,   8,
1470      8,   8,   8,   8,   8,   8,   3,   8,   7,   8,   8}
1480 m_vset(84,v)
1490 /*
1500 v={
1510 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1520     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1530 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1540     31,   0,   0,   0,   0,  25,   0,   2,   3,   0,   0,
1550     21,  15,   0,   5,   5,   0,   0,   1,   3,   0,   0,
1560     31,   0,   0,   0,   0,  30,   0,   8,   7,   0,   0,
1570     31,  15,   0,   5,   5,   3,   0,   4,   7,   0,   0}
1580 m_vset(85,v)
1590 /*
1600 v={
1610 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1620     44,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1630 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1640     31,   0,   0,   0,   0,  22,   0,   8,   3,   0,   0,
1650     18,   0,   0,   6,   0,   2,   0,   8,   3,   0,   0,
1660     31,   0,   0,   0,   0,  22,   0,   8,   7,   0,   0,
1670     31,   0,   0,   6,   0,   2,   0,   8,   7,   0,   0}
1680 m_vset(86,v)
1690 /*
1700 v={
1710 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1720     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1730 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1740     16,   0,   0,   7,   0,  36,   0,   1,   7,   0,   0,
1750     16,   0,   0,   7,   0,  46,   0,   8,   3,   0,   0,
1760     16,   0,   0,   7,   0,  46,   0,   1,   7,   0,   0,
1770     16,   6,   0,   7,   1,   0,   0,   2,   3,   0,   0}
1780 m_vset(87,v)
1790 /*
1800 v={
1810 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1820     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1830 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1840     31,   0,   0,  13,   0,  26,   0,   4,   7,   0,   0,
1850     14,  15,   0,  13,   2,   2,   0,   2,   7,   0,   0,
1860     31,   0,   0,  13,   0,  31,   0,   8,   3,   0,   0,
1870     14,  15,   0,  13,   2,  10,   0,   4,   3,   0,   0}
1880 m_vset(88,v)
1890 /*
1900 v={
1910 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1920     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1930 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1940     31,   0,   0,   0,   0,  26,   0,   8,   7,   0,   0,
1950     21,  16,   0,   7,   5,   0,   0,   4,   7,   0,   0,
1960     31,   0,   0,   0,   0,  31,   0,   4,   3,   0,   0,
1970     21,  16,   0,   7,   5,   8,   0,   2,   3,   0,   0}
1980 m_vset(89,v)
1990 /*
2000 v={
2010 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
2020     58,  15,   3,   1, 237,   0,  88,   0,   1,   3,   0,
2030 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
2040     28,   4,   0,   2,   1,  38,   2,   1,   3,   0,   0,
2050     22,   9,   1,   0,   1,  48,   2,  12,   0,   0,   0,
2060     29,   4,   3,   3,   1,  38,   1,   3,   7,   0,   0,
2070     15,   7,   4,   2,   5,   0,   2,   1,   0,   0,   0}
2080 m_vset(90,v)
2090 /*
2100 m_init():for i=1 to 8:m_alloc(i,8000):next
2110 for i=1 to 8:m_assign(i,i):next
2120 str a[256],b[256],c[256],d[256],e[256],f[256]
2130 str g[256],h[256],j[256],k[256],l[256],m[256]
2140 str n[256],o[256],p[256],q[256],r[256],s[256]
2150 str t[256],u[256],w[256],x[256],y[256],z[256]
2160 str aa[256],bb[256],cc[256],dd[256],ee[256],ff[256]
2170 str gg[256],hh[256],jj[256],kk[256],ll[256],mm[256]
2180 str nn[256],oo[256],pp[256],qq[256],rr[256],ss[256]
2190 str tt[256],uu[256],ww[256],xx[256],yy[256],zz[256]
2200 str sd[256],bd[256],t1[256],t2[256],t3[256],t4[256]
2210 key 19,"m_tempo(200)"+chr$(13):key 20,"m_tempo(117)"+chr$(
13)
2220 /*
2230 /*
2240 a="t117  [d.c.] o5 @l1r116  @71 p3 q8    v14 y48,20 y15,0
2250 b="eeeev13plep2ep3v14eeeev13p2eplfrv14p3fee
2260 c="ddddv13pldp2dp3v14ddddv13p2dplerv14p3edd
2270 c=b+c+c+b
2280 d="@78@v0@79p3o2|:4r@78@v0@79v14>gv8plr8@78@v0@79v14gv8plr
8@78@v0@79v14g6...v8plr4r@78@v0@79v14gv8plr8@78@v0@79v14gv8plr8@
78@v0@79v14g6...v8pl
2290 e="r4r<@78@v0@79v14cv8plr8@78@v0@79v14cv8plr8@78@v0@79v14c
6...v8plr4r@78@v0@79v14cv8plr8@78@v0@79v14cv8plr8@78@v0@79v14c6.
..v8plr4:|
2300 f="@85o5v11|:3r8aa<e>a<a>ar2r8gg<d>g<g>gr2r8aa<e>a<a>ar2r8
aa<e>a<a>ar8e8dc8d:|r8aa<e>a<a>ar2r8gg<d>g<g>gr2r8aa<e>a<a>ar2rl
o5@71p3q8v14y48,20
2310 zz="[*]
2320 m_trk(1,a) /* OPMA の SamplingFile
2330 m_trk(1,c) /* に あわせて ちょうせつしてあります。
2340 m_trk(1,"[coda]"+c) /*
2350 m_trk(1,c) /*なるへ゛くOPMAて゛きいてくた゛さい。
2360 m_trk(1,c) /*
2370 m_trk(1,c) /*
2380 m_trk(1,c) /*
2390 m_trk(1,c) /*
2400 m_trk(1,d) /*
2410 m_trk(1,e) /*
2420 m_trk(1,f) /*
2430 m_trk(1,zz) /*
2440 /*
2450 /*
2460 a="      [d.c.] o5 @l1r116  @71 p3 q8    v14 y49,16
2470 b="ccccv13plcp2cp3v14ccccv13p2cplcp2rv14p3ccc
2480 c=">bbbbv13plbp2bp3v14bbbbv13p2bplbp2rv14p3bbb<
2490 c=b+c+c+b
2500 d="@78@v0@79p2o2|:4r@78@v0@79v14>p2ev8plr8@78@v0@79v14p2ev
8plr8@78@v0@79p2v14e6...v8plr4r@78@v0@79v14p2ev8plr8@78@v0@79v14
p2ev8plr8@78@v0@79v14p2e6...v8pl
2510 e="r4r@78@v0@79v14p2av8plr8@78@v0@79v14p2av8plr8@78@v0@79v
14p2a6...v8plr4r@78@v0@79v14p2av8plr8@78@v0@79v14p2av8plr8@78@v0
@79v14p2a6...v8plr4:|
2520 f="@85o5v11y49,44|:3r8aa<e>a<a>ar2r8gg<d>g<g>gr2r8aa<e>a<a
>ar2r8aa<e>a<a>ar8e8dc8d:|r8aa<e>a<a>ar2r8gg<d>g<g>gr2r8aa<e>a<a
>ar2rlo5@71p3q8v14y49,16
2530 m_trk(2,a) /*
2540 m_trk(2,c) /*
2550 m_trk(2,"[coda]"+c) /*
2560 m_trk(2,c) /*
2570 m_trk(2,c) /*
```

▶いやはや，なんともありがたいディスクでした。あの福原氏のCGデータがあっただけでもうれしいのに，この充実ぶり。うーん，そういえば猫のCGを見て，昔常連になるぞ宣言をしたときの山田氏の絵とはだいぶ違う雰囲気がしました。　　富田　祐樹(18)東京都

```
2580 m_trk(2,c) /*
2590 m_trk(2,c) /*
2600 m_trk(2,c) /*
2610 m_trk(2,d) /*
2620 m_trk(2,e) /*
2630 m_trk(2,f) /*
2640 m_trk(2,zz) /*
2650 /*
2660 /*
2670 a="       [d.c.] o4 @11r116  @71 p3 q8     v14 y50,24
2680 b="bbbbv13p1ap2ap3v14aaaav13p2ap1ap3rv14p3aaa
2690 c="aaaav13p1gp2gp3v14ggggv13p2gp1gp3rv14p3ggg
2700 c=b+c+c+b
2710 d="@78@v0@80v13o5|:15v9p1cv5p2cv13p3ccv9p1cv5p2cv13p3ccv9p
1cv13p3cp2cp3cp1cp3cv9p2cv13p3c:|
2720 e="v9p1cv5p2cv13p3ccv9p1cv5p2cv13p3cc
2730 f="@77o5@v125d4@v126c4@v127d8.d8.c8d8c>a8.<e8d4d8c8d8.>g8.
a3...r4g8ae8.r2<erdrcrd8r8d8c8d8c>a8.<e8d4d8c8d8.>g8.a3...v11bgd
a4r8.@83o5v15q7p2arerg<crc>bragrap3
2740 g="@v127@77o5d8.d8.c8d8c>a8.<e8d4d8c8d8.>g8.a3...r4g8ae8.r
2<erdrcrd8r8d8c8d8c>a8.<e8d4d8c8d8.>g8.a3...r4ergar1o4@71p3q8v14
y50,24
2750 m_trk(3,a) /*
2760 m_trk(3,c) /*
2770 m_trk(3,"[coda]"+c) /*
2780 m_trk(3,c) /*
2790 m_trk(3,c) /*
2800 m_trk(3,c) /*
2810 m_trk(3,c) /*
2820 m_trk(3,c) /*
2830 m_trk(3,d) /*
2840 m_trk(3,e) /*
2850 m_trk(3,f) /*
2860 m_trk(3,g) /*
2870 m_trk(3,zz) /*
2880 /*
2890 /*
2900 a="       [d.c.] o4 @12r116  @88 p3 q8 r16 v8 y51,20
2910 b="a2..e8g2..r8g2..b8a2..r16[coda]o1@72
2920 c="|:6@v127a8v7p1rp2rp3@v127egaav7p2rp3@v127av7p1rp3@v127e
av7p2rp3<@v127cv7p1rp3>@v127g8v7p2rp1rp3@v127deggv7p1rp3@v127gv7
p2rp3@v127dgv7p1rp3@v127dv7p2rp3@v127g8v7p1rp2rp3
2930 d="@v127deggv7p2rp3@v127gv7p1rp3@v127dgv7p2rp3@v127dv7p1rp
3@v127a8v7p2rp1rp3@v127egaav7p1rp3@v127av7p2rp3@v127eav7p1rp3@v1
27ev7p2rp3:|
2940 e="|:4@v127g8v7p1rp2rp3@v127>gg<degg<g>gv7p2rp1rp3@v127dv4
p2rp3@v127g8v7p1rp2rp3>@v127gg<degg<g>gv7p1rp3@v127eg8
2950 f="a8v7p2rp1rp3@v127>aa<egaa<a>av7p2rp1rp3@v127ev7p1rp3@v1
27a8v7p2rp1rp3>@v127aa<egaa<a>av7p2rp3@v127ae8:||:3
2960 g="f8v7p1rp2rp3@v127fcdfv7p1rp3@v127fv7p2rp3@v127cfv7p1rp3
@v127cv7p2rp3@v127e8v7p1rp2rp3@v127eb<e>ev7p1rp3@v127ev7p2rp3
2970 h="@v127dev7p1rp3@v127gv7p2rp3@v127a8v7p1rp2rp3@v127aegav7
p1rp3@v127av7p2rp3@v127gab<cv7p1rp3>@v127a8v7p2rp1rp3@v127aegav7
p2rp3@v127av7p1rp3@v127gab<cv7p2rp3>@v127:|
2980 j="@v127dev7p1rp3@v127gv7p2rp3@v127a8v7p2rp1rp3@v127aegav7
p2rp3@v127av7p1rp3@v127gav7p2rp3@v127aav7r1
2990 m_trk(4,a) /*
3000 m_trk(4,b) /*
3010 m_trk(4,c) /*
3020 m_trk(4,d) /*
3030 m_trk(4,e) /*
3040 m_trk(4,f) /*
3050 m_trk(4,g) /*
3060 m_trk(4,h) /*
3070 m_trk(4,g) /*
3080 m_trk(4,j) /*
3090 m_trk(4,zz) /*
3100 /*
3110 /*
3120 a="       [d.c.] o4 @12r 18  @88 p2 q8  r8 v8 y52,12
3130 b="a2..eg2..rg2..@78@v0o2@73p2v14a&v13a&v12a&v12a&v11a&v10
a&v9a&v8a[coda]@78@v0@76v13o6q5aq8@89p1v13o3116
3140 c="arerarearaerarergrdrgrdgrgdrgrv10grv13grdrgrdgrgdrgrbra
rerarearaera@78@v0r@76v13o6q5a8q8@89p1v13o3
3150 d="arerarearaerarergrdrgrdgrgdrgrv10grv13grdrgrdgrgdrgr@76
v10o4q4p1ap3b<p2c>p3b<p1cp3dp2ep3degab<cgeg @78@v0@76v13o6q5ar4.
.
3160 e="@77o4@v127q8a8.b8.<c8>b4.g3...r2g8.a8.b8b4.a3...@76v13o
6q5ar4..@77o4@v127q8a8.b8.<c8>b4.g3...r2g8.g8.<d8c4.>a2&a@78@v0r
v13@76o6q5ar4..
3170 f="@77o5@v127q8c8.>b8.a8b8.g8.<d3...d8.d8.d8d8.c8.>b8a8.b8
.a2&a@78@v0rv13@76o6q5ar4..
3180 g="@77o5@v127q8c8.>b8.a8b8.g8.<d3...>b8.b8.b8b8.a8.g8a2r2@
78@v0@11r116
3190 h="@82p1v12o5q7|:2drergrdergrdergr:||:2ergraregraregrar:||
:2drergrdergrdergr:|
3200 j="@147r116@83q7p3v15o5arga8.<cr8.>p2v12gap1v10gar8p3v14ar
g<drdcr4r@11r116
3210 k="@84v12p2o3ergraregraregrarergrargar4@147r116
3220 l="@86v7o1a1g1e1a4b4<c4d4>a1g1e2.@83o5v13q7r4.arerg<crc>br
agra@86v7o1a1g1e1@87p3o6v9c4&c>b4a4g4a1r8drergargrge-&dc>ar<dc>a
2@83p1o5@v127q7ergar1y52,12
3230 m_trk(5,a) /*
3240 m_trk(5,b) /*
3250 m_trk(5,c) /*
3260 m_trk(5,d) /*
3270 m_trk(5,e) /*
3280 m_trk(5,f) /*
3290 m_trk(5,g) /*
3300 m_trk(5,h) /*
3310 m_trk(5,j) /*
3320 m_trk(5,h) /*
3330 m_trk(5,k) /*
3340 m_trk(5,l) /*
3350 m_trk(5,zz) /*
3360 /*
3370 /*
3380 a="       [d.c.] o4 @12r 18  @88 p1 q8  r8 v8 y53,28
3390 b="a2..eg2..rg2..@78@v0o2@73p1v8a&v9a&v10a&v11a&v12a&v12a&
v13a&v14a[coda]@78@v0@76v13o5q5aq8@89p2v13o3116
3400 c="arerarearaerarergrdrgrdgrgdrgrv10grv13grdrgrdgrgdrgrbra
rerarearaera@78@v0r@76v13o5q5a8q8@89p2v13o3
3410 d="arerarearaerarergrdrgrdgrgdrgrv10grv13grdrgrdgrgdrg@78@
v0r@76v10o3q4p3ap1b<p3c>p2b<p3cp1dp3ep2dp3egab<cgeg @78@v0@76v13
o5q5ar4..r8
3420 e="@90o4v12q8a8.b8.<c8>b4.g3...r2g8.a8.b8b4.a2@76v13o5q5ar
4..r8@77o4v12q8a8.b8.<c8>b4.g3...r2g8.g8.<d8c4.>a4..@78@v0rv13@7
6o5q5ar4..r8
3430 f="@90o5v12q8c8.>b8.a8b8.g8.<d3...d8.d8.d8d8.c8.>b8a8.b8.a
4..@78@v0rv13@76o5q5ar4..r8
3440 g="@90o5v12q8c8.>b8.a8b8.g8.<d3...>b8.b8.b8b8.a8.g8a2r4.@7
8@v0
3450 h="@82p2v12o5q7|:2drergrdergrdergr:||:2ergraregraregrar:||
:2drergrdergrdergr:|
3460 j="r4@83q7p3v15o4y53,36arga8.<cr8.>p2v12gap1v10gar8p3v14ar
g<drdcr4ry53,28
3470 k="@84v5p3o3rergraregraregrarergrargar4..
3480 l="y53,48@86o1v7|:2a1g1e1a4b4<c4d4>:|a1g1e1@87p1o6v12q7c4>
b4a4g4a1r8drergargrge-&dc>ar<dc>a2&a<crder1y53,28
3490 m_trk(6,a) /*
3500 m_trk(6,b) /*
3510 m_trk(6,c) /*
3520 m_trk(6,d) /*
3530 m_trk(6,e) /*
3540 m_trk(6,f) /*
3550 m_trk(6,g) /*
3560 m_trk(6,h) /*
3570 m_trk(6,j) /*
3580 m_trk(6,h) /*
3590 m_trk(6,k) /*
3600 m_trk(6,l) /*
3610 m_trk(6,zz) /*
3620 /*
3630 /*
3640 a="       [d.c.] o4 @12r 18  @74 p3 q8     v9 y54,00
3650 b="a2..eg2..rg2..ba1116>
3660 c="[coda]@75@v127o3arerarearaerarergrdrgrdgrgdrgv12rgrv15g
rdrgrdgrgdrgrbrarerarearaerar8.
3670 d="r2@90p1o4v10q8r4a8.b8.<c8>b4.g3...r2g8.a8.b8b4.a1&a8a8.
b8.<c8>b4.g3...r2g8.g8.<d8c4@75v13a<a>ar v11a<a>arr2
3680 e="@90p1o5v10q8r4c8.>b8.a8b8.g8.<d3...d8.d8.d8d8.c8.>b8a8.
b8.a4.r2.
3690 f="@90p1o5v10q8c8.>b8.a8b8.g8.<d3...>b8.b8.b8b8.a8.g8a4..r
32.@78@v0r64
3700 g="@81@v127o3a8.g&g1&g2e4g8.a8er8er8.r1r4a8.g1&g2&gd4g8.e8
e-c>ar2.r2.<a8.g&g1&g2
3710 h="e4g8.a8er8er8.r1r4a8.g1&g2&ge4g8.ar1r1
3720 j="@86v7o1c1>b1a1a4b4<c4d4c1>b1a1a8@83o4@v127q7arerg<crc>b
ragra@86v7o1c1>b1a1a4b4<c4d4c1>b1a2.@83o4@v127arb<cr1y54,00
3730 m_trk(7,a) /*
3740 m_trk(7,b) /*
3750 m_trk(7,c) /*
3760 m_trk(7,c) /*
3770 m_trk(7,d) /*
3780 m_trk(7,e) /*
3790 m_trk(7,f) /*
3800 m_trk(7,g) /*
3810 m_trk(7,h) /*
3820 m_trk(7,j) /*
3830 m_trk(7,zz)/*
3840 /*
3850 /*
3860 a="       [d.c.] o4 @12r 18  @74 p3 q8     v9 y55,40 y3,3
3870 b="a2..eg2..rg2..ba2&116y2,23a&y2,23a&y2,23a&y2,23a&y2,23a
&y2,23a&y2,23a&y2,23a>
3880 c="[coda]@75@v127y3,3y2,50arery2,14ay3,2y2,8ry3,3y2,23ey2,
23ary2,23aery2,14ary3,1y2,66ery3,3y2,23grdry2,14gy3,2y2,8ry3,3y2
,23dy2,23gry2,23gdry2,14gv12ry3,1y2,66gry3,3y2,23v15grdr
3890 d="y2,14gy3,2y2,8ry3,3y2,23dy2,23gry2,23gdry2,14gry3,1y2,6
6bry3,3y2,23arery2,14ay3,2y2,8ry3,3y2,23ey2,23ary2,23aery2,14ary
2,14ry2,14r
3900 e="y2,14gy3,2y2,8ry3,3y2,23dy2,23gry2,23gdry2,14gry3,1y2,6
6bry3,3y2,23arery2,14ay3,2y2,8ry3,3y2,23ey2,23ary2,23aery2,14ay2
,14ry2,14ry2,14r
3910 f="y2,50r4y2,14ry3,2y2,8ry3,3y2,23ry2,23r@90p2o4v10q8ry2,2
3r8.y2,14a8&y3,1y2,66ab&y3,3y2,23b8<c8>y2,14b&y3,2y2,8b&y3,3y2,2
3b&y2,23b8&y2,23b
3920 g="g8&y2,14g8&y3,1y2,66g8&y3,3y2,23g4y2,14ry3,2y2,8ry3,3y2
,23ry2,23r8y2,23r8.y2,14g8&y3,1y2,66ga&y3,3y2,23a8b8y2,14b&y3,2y
2,8b&y3,3y2,23b&y2,23b8&y2,23ba8&y2,14a8&y2,14a&y2,14a&y2,50a4&
3930 h="y2,14a&y3,2y2,8a&y3,3y2,23a&y2,23a8&y2,23a8.y2,14a8&y3,
1y2,66ab&y3,3y2,23b8<c8>y2,14b&y3,2y2,8b&y3,3y2,23b&y2,23b8&y2,2
3bg8&y2,14g8&y3,1y2,66
3940 j="g8&y3,3y2,23g4y2,14ry3,2y2,8ry3,3y2,23ry2,23r8y2,23r8.y
2,14g8&y3,1y2,66gg&y3,3y2,23g8<d8y2,14c&y3,2y2,8c&y3,3y2,23c&y2,
23c@75v13>a<y2,23a>arv11y2,14a<a>y2,14ay2,14ry2,50r4y2,14ry3,2y2
,8ry3,3y2,23ry2,23r8
3950 k="@90p2o5v10q8y2,23r8.y2,14c8&y3,1y2,66c>b&y3,3y2,23b8a8y
2,14b&y3,2y2,8b&y3,3y2,23by2,23g8&y2,23g<d8&y2,14d8&y3,1y2,66d8&
y3,3y2,23d4y2,14d&y3,2y2,8d&y3,3y2,23dy2,23d8&y2,23dd8y2,14d8&y3
,1y2,66dc&y3,3y2,23c8>b8y2,14a&y3,2y2,8a&y3,3y2,23ay2,23b8
3960 l="&y2,23ba8&y2,14a8&y2,14a&y2,14ay2,50r4y2,14ry3,2y2,8ry3
,3y2,23ry2,23r8y2,23r8.
3970 m="@90p2o5v10q8y2,14c8&y3,1y2,66c>b&y3,3y2,23b8a8y2,14b&y3
,2y2,8b&y3,3y2,23by2,23g8&y2,23g<d8&y2,14d8&y3,1y2,66d8&y3,3y2,2
3d4
3980 n=">y2,14b&y3,2y2,8b&y3,3y2,23by2,23b8&y2,23bb8y2,14b8&y3,
1y2,66ba&y3,3y2,23a8g8y2,14a&y3,2y2,8a&y3,3y2,23a&y2,23a8&y2,23a
8&a32.@78@v0r64
3990 o="@81@v127y55,48o3y2,14a&y2,14a&y2,14ay2,14g&y2,23g4&y2,1
4g&y3,2y2,8g&y3,3y2,23g&y2,23g8&y2,23g8.&y2,14g8&y3,1y2,66g8&
4000 p="y3,3y2,23g4&y2,14g&y3,2y2,8g&y3,3y2,23g&y2,23ge&y2,23e8
.y2,14g&y3,1y2,8g&y3,2y2,8ga&y3,3y2,23aer8y2,14ey3,2y2,8ry3,3y2,
23ry2,23r8y2,23r8.y2,14r8y3,1y2,66r8y3,3y2,23r4y2,14ry3,2y2,8ry3
,3y2,23ry2,23r8y2,23r8.
4010 q="y2,14a&y3,1y2,8a&y3,2y2,8ag&y3,3y2,23g4&y2,14g&y3,2y2,8
g&y3,3y2,23g&y2,23g8&y2,23g8.&y2,14g8&y3,1y2,66g8&y3,3y2,23g4&y2
,14g&y3,2y2,8g&y3,3y2,23g&y2,23gd&y2,23d8.
4020 r="y2,14g&y3,1y2,8g&y3,2y2,8ge&y3,3y2,23ee-c>ay2,14ry3,2y2
,8ry3,3y2,23ry2,23r8y2,23r8.y2,14r8y3,1y2,66r8y3,3y2,23r4y2,14ry
3,2y2,8ry3,3y2,23ry2,23r8y2,23r8.
4030 s="<y2,14a&y2,14a&y2,14ay2,14g&y2,23g4&y2,14g&y3,2y2,8g&y3
```

▶やっと出ましたね，X68000マシン語プログラミング＜入門編＞。あまりのうれしさにまだ全部読み終わっていないのに，愛読者カードを出してしまった。青木　俊一(21)東京都

```
,3y2,23g&y2,23g8&y2,23g8.&y2,14g8&y3,1y2,66g8&y3,3y2,23g4&y2,14g
&y3,2y2,8g&y3,3y2,23g&y2,23ge&y2,23e8.
 4040 t="y2,14g&y3,1y2,8g&y3,2y2,8ga&y3,3y2,23aer8y2,14ey3,2y2,8
ry3,3y2,23ry2,23r8y2,23r8.y2,14r8y3,1y2,66r8y3,3y2,23r4y2,14ry3,
2y2,8ry3,3y2,23ry2,23r8y2,23r8.
 4050 u="y2,14a&y3,1y2,8a&y3,2y2,8ag&y3,3y2,23g4&y2,14g&y3,2y2,8
g&y3,3y2,23g&y2,23g8&y2,23g8.&y2,14g8&y3,1y2,66g8&y3,3y2,23g4&y2
,14g&y3,2y2,8g&y3,3y2,23g&y2,23ge&y2,23e8.y2,14g&y3,1y2,8g&y3,2y
2,8ga
 4060 w="y3,3y2,23ry3,1y2,32r32y2,32r32y2,32ry3,3y2,32ry2,14ry3,
2y2,8ry3,3y2,23ry2,34r32y2,34r32y2,34ry2,34ry3,2y2,35rry3,3y2,14
rry3,1y2,66rry3,3y2,23ry3,1y2,32r32y2,32r32y2,32ry3,3y2,32ry2,14
ry2,34ry3,1y2,32ry3,3y2,23rr
 4070 x="y2,16r32y2,16r32y2,15ry2,15ry2,15ry2,14ry2,14ry2,14r@86
o1v7|:3
 4080 y="y2,23e8&y2,23e8&y2,14e&y3,2y2,8e&y3,3y2,23e&y2,23e8&y2,
23e8.&y2,14e8&y3,1y2,66e8y3,3
 4090 z="y2,23d8&y2,23d8&y2,14d&y3,2y2,8d&y3,3y2,23d&y2,23d8&y2,
23d8.&y2,14d8&y3,1y2,66d8y3,3
 4100 aa="y2,23c8&y2,23c8&y2,14c&y3,2y2,8c&y3,3y2,23c&y2,23c8&y2
,23c8.&y2,14c8&y3,1y2,66c8y3,3
 4110 bb=">y2,23a8&y2,23a8y2,14b&y3,2y2,8b&y3,3y2,23b&y2,23b<c&y
2,23c8.y2,14d8&y2,14d&y2,14d:|
 4120 cc="y2,23c8&y2,23c8&y2,14c&y3,2y2,8c&y3,3y2,23c&y2,23c8&y2
,23c8.y2,14@83@v127o5ery2,14gy2,14ar2.y2,23ry2,23ry2,23ry2,23r
 4130 m_trk(8,a) /*
 4140 m_trk(8,b) /*
 4150 m_trk(8,c) /*
 4160 m_trk(8,d) /*
 4170 m_trk(8,c) /*
 4180 m_trk(8,e) /*
 4190 m_trk(8,f) /*
 4200 m_trk(8,g) /*
 4210 m_trk(8,h) /*
 4220 m_trk(8,j) /*
 4230 m_trk(8,k) /*
 4240 m_trk(8,l) /*
 4250 m_trk(8,m) /*
 4260 m_trk(8,n) /*
 4270 m_trk(8,o) /*
 4280 m_trk(8,p) /*
 4290 m_trk(8,q) /*
 4300 m_trk(8,r) /*
 4310 m_trk(8,s) /*
 4320 m_trk(8,t) /*
 4330 m_trk(8,u) /*
 4340 m_trk(8,w) /*
 4350 m_trk(8,x) /*
 4360 m_trk(8,y) /*
 4370 m_trk(8,y) /*
 4380 m_trk(8,aa) /*
 4390 m_trk(8,bb) /*
 4400 m_trk(8,y) /*
 4410 m_trk(8,z) /*
 4420 m_trk(8,cc) /*
 4430 m_trk(8,zz) /*
 4440 /*
 4450 /*
 4460 m_play()
```

日本音楽著作権協会(出)許諾第9072264-001号

リスト2 スプーンおばさん

```
10 '
20 '
30 '
40 '    RINGONOMORI
50 '
60 '
70 '
80 '
90 '
100 INIT:DEFSTR A-Z:DEFINT X:GOSUB 1400
110 DIM A(16),B(16),C(16),D(16),E(16),F(16),G(16)
120 TEMPO0
130 A(0)="O6V12Q7L8I1
140 B(0)="O5V12Q7L8I1
150 C(0)="O2V13Q7L8I1
160 D(0)="O4V11Q7L8I1
170 E(0)="O4V11Q7L8I1
180 F(0)="O4V12Q7L8I1
190 G(0)="O5V6I3Q3L8
200 A(1)="C<B-AGFEDC< B-AGFECDE GFEFGFEF
210 B(1)="R1 R1 R1
220 C(1)="C>C<G>C<C>C<G>C <C>C<G>C<CCDE F4C4F4D-4
230 D(1)="R1 R1 R1
240 E(1)="R1 R1 R1
250 F(1)="R1 R1 R1
260 G(1)="R1 R1 R1
270 A(2)="B-4.ARAAA AFFDD<B-B-G> <GA>DCRCDE
280 B(2)="R1 R1 R1
290 C(2)="D4G-4A4D4 G4D4G4D4 C4G4CCDE
300 D(2)="R1 R1 R1
310 E(2)="R1 R1 R1
320 F(2)="R1 R1 R1
330 G(2)="V12R4R4.CCC R1 R4R4.CCC
340 A(3)="GFEFGFEF B-4.AR>DDD
350 B(3)="R1 R1
360 C(3)="F4C4F4D-4 D4G-4A4D4
370 D(3)="R1 R1
380 E(3)="R1 R1
390 F(3)="R1 R1
400 G(3)="R1 R4R4.CCCV6
410 A(4)="D<AAFFDFA E4EFR2
420 B(4)="R1 C4<ARR2
430 C(4)="G4D4G4D4 C4CFRCC4
440 D(4)="R1 R1
450 E(4)="R1 R1
460 F(4)="R1 R1
470 G(4)="R1 R1
480 A(5)="<A4>C4G4.F8 EFC<A>C<AB->C C<B-B-AAGB->D
490 B(5)="R1 R1 R1
500 C(5)="F4C4F4C4 F4F4G-4G-4 G4D4G4D4
510 D(5)="R1 R1 R1
520 E(5)="R1 R1 R1
530 F(5)="R1 R1 R1
540 G(5)="I2O2V11Q8F2F2 F2F#2 G2G2
550 A(6)="G2.R4 <B-4>D4A4.G8 G-GD<B->D<B->CD
560 B(6)="R1 R1 R1
570 C(6)="G4D4GFED G4D4G4D4 G4D4G4D4
580 D(6)="R1 R1 R1
590 E(6)="R1 R1 R1
600 F(6)="R1 R1 R1
610 G(6)="G2&G<B->DG B-2A2 G-2G2
620 A(7)="EDDCCDEC A2R4C4 >C4.<B-B-A4B-
630 B(7)="R1 R1 R1
640 C(7)="C4G4C4G4 F4C4F4FG A4E-4D4G-4
650 D(7)="R1 R1 R1
660 E(7)="R1 R1 R1
670 F(7)="R1 R1 R1
680 G(7)="C2C2 V6F2.FG A2B-2
690 A(8)="AGG-GB-2 DERDERDE DERDE2& E2RCDE
700 B(8)="R1 R1 FGRFG2& G2R2
710 C(8)="G4D4G4GA B->CR<B->CR<B->C <B->CR<B->C2 C<B-GECCDE
720 D(8)="R1 R1 R1 R1
730 E(8)="R1 R1 R1 R1
740 F(8)="R1 R1 R1 R1
750 G(8)="A2G2 DERDERDE R1 C<B-GECCDE
760 A(9)="GFEFGFEF B-4.ARAAA AFFDD<B-B-G
770 B(9)="R1 R1 R1
780 C(9)="F4C4F4D-4 D4G-4A4D4 G4D4G4D4
790 D(9)="R1 R1 R1
800 E(9)="R1 R1 R1
810 F(9)="R1 R1 R1
820 G(9)="I3Q3O5V12R1 R4R4.CCC R1V6
830 A(10)="GA>DCRCDE GFEFGFEF B-4.AR>DDD D<AAFFDFA
840 B(10)="R1 R1 R1 R1
850 C(10)="C4G4CCDE F4C4F4D-4 D4G-4A4D4 G4D4G4D4
860 D(10)="R1 R1 R1 R1
870 E(10)="R1 R1 R1 R1
880 F(10)="R1 R1 R1 R1
890 G(10)="R4R4.CCC R1 R1 R1
900 A(11)="E4EFR2 >D4F4G4FD CF4A&A2
910 B(11)=">C4<ARR2 R1 R1
920 C(11)="C4CFRFGA B-4F4B-4B-4 F4C4F4GA
930 D(11)="R1 R1 R1
940 E(11)="R1 R1 R1
950 F(11)="R1 R1 R1
960 G(11)="I1O2Q8R1 F2F4GA C2C2
970 A(12)="D4F4G4FG <B4>V13FV10<B-&B-2 R<AAAAA4>>>C16&D16
980 B(12)="R1 D4RD&D2< RCCCCC4R
990 C(12)="B-4F4B-4F4 >A4RV13G&G2V11 RFFFEE4R
1000 D(12)="D2D-2 <G4.G4AB-G >C1&
1010 E(12)="R1 R1 R1
1020 F(12)="R1 A4AG&G2 R1<<
1030 G(12)="R1 R1 R1
1040 A(13)="GFEDC<AEC R<<B-B-B-B-B-4>>G16&A16 B-AGFD-C<B-F
1050 B(13)="R1 R1 R1
1060 C(13)="R1 RGGGFF4R R1
1070 D(13)="C1 <B-1& B-1>>
1080 E(13)="R1 R1 R1
1090 F(13)="R1 R1 R1
1100 G(13)="R1 R1 R1
1110 A(14)="RDD4DDR>D D4DDR2V12<
1120 B(14)="RFF4FFR>F F4FFR2<
1130 C(14)="RB-B-4B-B-R>B- B-4B-B-R2<<
1140 D(14)="R<CC4CCRC C4CCR<CDE>>
1150 E(14)="R1 R1
1160 F(14)="R1 R1
1170 G(14)="R1 R1
1180 A(15)="E4EFRCDE
1190 B(15)="V13C4<ARR2V10
1200 C(15)="R1
1210 D(15)="V13C4CFRV4CDEV10
1220 E(15)="R1
1230 F(15)="R1
1240 G(15)="R1
1250 A(16)="O4E4EFR>DDD D<AAFFDFA E4EFR<GA>C F4R<<<A16A16A4R4
1260 B(16)="O4V12C4ARR2 R1 C4ARR2 R4RC16C16C4R4
1270 C(16)="R1 R1 R1 R4RF16F16F4R4
1280 D(16)="R1 R1 R1 R4R<<F16F16F4R4
1290 E(16)="R1 R1 R1 R1
1300 F(16)="R1 R1 R1 R4RA16A16A4R4
1310 G(16)="R1 R1 R1 R1
1320 PLAY A(0)+":"+B(0)+":"+C(0)+":"+D(0)+":"+E(0)+":"+F(0)+":"+
G(0)
1330 RESTORE
1340 READ X:IF X=999 THEN END
1350 PLAY A(X)+":"+B(X)+":"+C(X)+":"+D(X)+":"+E(X)+":"+F(X)+":"+
G(X)
1360 GOTO 1340
1370 DATA 1,2,3,4,5,6,7,8,9,10,11,12
1380 DATA 13,14,5,6,7,8,9,10,15,9,10,16
1390 DATA 999
1400 MEM$(&HB190,36)=HEXCHR$("FA 00 31 4C 33 51 25 2F 2F 00 9C 9
6 5D 8F 04 09 04 87 00 01 03 00 15 12 16 A5 80 80 80 80 00 DC 80
 00 02 80")'PIANO
1410 MEM$(&HB1B4,36)=HEXCHR$("F8 10 3F 36 31 51 31 23 2D 00 52 5
B 50 8E 9F 88 8A 80 14 0C 01 01 2A 2A 03 0A 81 81 81 82 F4 C8 D0
 00 02 80")'CELLO
1420 MEM$(&HB1D8,36)=HEXCHR$("E4 40 0C 0C 4C 3C 37 00 37 00 1A 1
3 12 14 95 97 83 8D 09 12 00 12 0C FB 04 FF 00 00 00 80 00 8B E0
 40 02 80")'BIRD
1430 RETURN
```

ノスタルジア
噂ばかりが先行していたが，ようやく姿を現したぞ。いままでのアドベンチャーとはひと味もふた味も違う作りだ。

SOFTWARE INFORMATION

今月もいろいろな情報が飛び込んできました。いやあ，この時期はいいですねえ，ネタに困らなくって。でもX68000を中心に動いているメーカーさんも増えてきたし，こういう状態がずっと続くといいなぁ。

話題のソフトウェア

さあ，どこもかしこも年末進行。アタマはみんなナチュラルハイ！ X68000に向かって笑いかけてる自分がコワイぞ，へへ。

まずは，タケルの**ノスタルジア**。豪華客船で突如発生した乗っ取り事件，偶然乗り合わせた主人公は事件解決に向けて行動を開始する……，てなアドベンチャー。でも，従来のそれとはちと違う。いわゆる“早解き”ふうではなく，会話を楽しみ“行動する”といったゲーム。主人公の感情や関心がそのままグラフィックで表されるし，時限爆弾解体シーンでも，ネジの回る様子を表示してスリル感をアップしている。こういった細かな感情の動きを絵にすることで，プレイヤーもどんどん引き込まれ，思わずニヤリってなわけです。感覚的には知的対話型アクションてカンジかな。

次は，システムサコムの**チャイム**。中学時代同級生だった３人の女の子が別々の高校へ進み，さまざまな人に出会い成長し，それぞれに道を歩んでいく様子を描いたアドベンチャー。登場人物はかなり多く，なかにはサコムの人をパロったものも。

そのほか電波新聞社では**エイリアンシンドローム**，SPSでは**メルヘンメイズ**，システムソフトでは**ボンバーマン**，ホビージャパンでは**ブラックレインボウ**と**ビーストロード**をそれぞれ開発中とのこと。それじゃ。

DOCTOR2.X使用上の注意

１月号の付録ディスクに収録されているウイルス検出用プログラム「DOCTOR2.X」は，フロッピーディスクに記録された IPL （イニシャル・プログラム・ローダー）という起動時に最初に読み込まれるプログラムをチェックしています。通常， IPL はHuman68kの純正のものがそのまま使われることが多いのですが，市販のゲームソフトにはオリジナルの IPL が使用される場合もあります。

そこで、DOCTOR2.Xを組み込んでいる場合，オリジナル IPL のゲームディスクを起動しようとすると，DOCTOR2.Xは，次のようなメッセージを出すようにしています。

1) これはHudson純正のIPLではないが大丈夫でしょう。シフトキーを押すと続行します。

また，一部のソフトで，オリジナル IPL を使用したものであるにもかかわらず次のようなメッセージが出ることがあります。

2) これは未知のウイルスに冒されているかもしれません。注射をしますので，ドライブ１にHuman Ver. 1.00（相当品）を入れて，リターンキーを押してください。

これは，本来Human純正のシステムが書き換えられている場合に出す警告ですが，DOCTOR2.Xの不備で，純正以外の IPL に対してもチェック項目が一致してしまったためです。もともと，書き込み不可のディスクではウイルスの侵入はないはずですから市販のソフトの場合はまず安全です。メッセージを見て驚かれるかもしれませんが，無理に治療しようとはしないでください。現在各ソフトハウスでは十分なチェックを行ってソフトを出荷しているはずです。

1)の場合はシフトキーを押せばそのまま起動しますが，2)の場合にはOPT.1キーを押しながらリセットして起動してください。

もちろん，通常お使いのシステムディスクで1)，2)のメッセージが出たときは注意が必要です。その場合には編集部までご連絡ください。

また，市販ソフトで1)または2)のメッセージが出た場合には，プロテクトシールを剝がさないように保存しましょう（最初から切り込みがないディスクは安全）。万が一ウイルスに感染するとオリジナル IPL が壊れてゲームは起動できなくなります。 （編集部）

リングマスターII

シナリオＩでリングナイトに昇格した主人公だったが，密命を帯び，ゴトランタ大陸のムルソン大公国を訪れる。フィリアス・ノギスに迫る戦争の予感。激動する事態を前に，君は使命をまっとうできるか……？

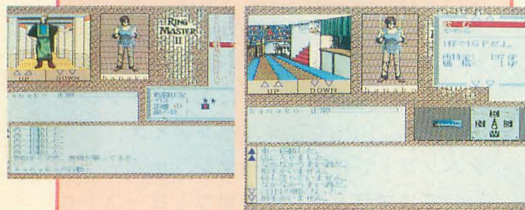
すいません。これ，ほとんどパッケージのコピーそのままです。しかしまあ，もともとシナリオが命のリングマスターですから，やりもしないうちに書いてしまうのもナニかなぁと。

というわけで，実際に遊んでのレポートはまた後日にするとして，ここでは概要だけ紹介することにしましょう。

リングマスターIIはテーブルトークの老舗ホビージャパンが，そのノウハウを生かして作り上げた本格派RPGです。スキル，メンタルポイント，さまざまな攻撃修正など，特にルール面がしっかりしているのが特徴。シナリオも綿密な世界観に基づいて書かれ，人口無脳を搭載した登場人物との会話によってストーリーを進めていくことができます。

X68000に関してはシナリオ，グラフィックなどによりきめ細かなチェックを入れ，より完成度を高めてあるとのことですから，RPGファンにはたまらない１本となることでしょう。

（ですます浦）

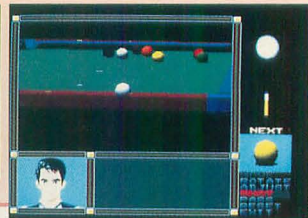
X68000用　5″2HD版3枚組　8,800円（税別）
ホビージャパン　☎03(3354)9341

Magical Shot

M.N.M Softwareのビリヤードゲーム「Magical Shot」もいよいよ発売を迎えたようだ。

ウリはなんといっても本格的な3D処理。ポリゴン処理を使ったビリヤードゲームは，たぶん日本では初めてじゃないかな。いろんな角度から台を眺められるというのは，ビリヤードでは重要なだけに嬉しい。マウスで台をスルスルと動かすのは，なかなか気分がいいぞ。さらに打ったあとにはインスタントリプレイも見られる。ナイスショットの快感を何度でも味わえるぞ。

ゲームはオーソドックスなナインボール。対戦相手は，アダルトな雰囲気の紳士淑女が８人揃っている。ビリヤードがお好きな人は必携の１本だぞ。

（ファンキー浦）

X68000用　5″2HD版　6,800円（税込）
ブラザー工業（TAKERU）　☎052(824)2493

ザークレジェンドスペシャル

ザーク レジェンド スペシャルは，新進ソフトハウス，マキシマのアクションゲームだ。高貴な血筋を引く主人公アンドレアが，大魔王ザークを倒すために立ち上がるというもの。ツボを押さえたストーリーですな。

ゲームは縦型のスクロール画面。現れる敵を剣や銃で倒し，入り組んだ地形をジャンプで克服してどんどん進んでゆくのだ。感じとしては，イースIIIなどよりは，むしろスーパーマリオとかビクターの「シャッタードフューチャー」に近い。みんな知らないか。

しかしマニュアルには「PC-9801でのソフトウェアスプライト」をウリにしているけど，X68000ではこれくらいの動きは並って感じがする。音楽も中村泰士氏ということだけど，ちょっとゲームを意識しすぎてかえって古いタイプのゲームミュージックだという気がする。

アクションゲームとして見ればアニメーションはきっちりしてるし，敵の配置なども考えてあるけど，そこはそれ，もはやゲーセンの移植を見ても驚かないX68000ユーザーが相手だけに厳しいかな。

（ファンキー浦）

X68000用　5″2HD版4枚組　8,800円（税別）
マキシマ　☎06(561)2215

スライス

M.N.M Softwareのパズルゲーム「スライス」がそろそろ登場の見込み。こちらには完成版が届いたので紹介しちゃおうかな。

まずルール。３個同じ種類のブロックを並べて消すという，コラムスと同じものだ。ううん，シンプルイズベスト。面白いのはブロックが着地してからで，ブロックを左右に倒すことができる。当然平坦なところばかりじゃないから，ひっくりかえったり，ブロックが分割したりする。さらにそれから移動させることもできるのだ。着地してからその先がどうなるか，なかなか予測がつかないけど，それだけに自分の思いどおりにいったり，思いがけず大量の連鎖反応を起こしてしまうと嬉しさに体が溶けてしまう。ああ，もうダメって感じ。

BGMはあの古代祐三氏。今回はスクラッチ技を使ったファンキーなBGMで楽しませてくれるぞ。“１，２，（ゴシュゴシュ）ヘイ！”って感じで。くーっ，イカすぜ。今日からみんなもスライスで，“１，２，（ゴシュゴシュ）ヘイ！”だ。ヘイ！

（ファンキー浦）

X68000用　5″2HD版　6,800円（税込）
ブラザー工業（TAKERU）　☎052(824)2493

大航海時代

光栄の大航海時代のX68000版が完成，発売された。このゲーム，船隊を率いて７つの海を渡り，国王に認められるほどの大人物になるのが目的というものなんだ。とにかく，みんなの評判になればなんでもヨシってのがいい。地道に貿易をして大商人になるもよし，海賊になって敵国の艦隊をドカドカやっつけるもよし，港に投資をして事業家になるもよし，自分の好きなようにプレイできるところが特徴なんだな。

X68000版はマウス専用オペレーション。昔よりだいぶ考えられていて，ウィンドウも好きな場所に動かせる。なかなか快適にプレイすることができるぞ。個人的には数字ぐらいキーボードで入れさせてほしかったけど。

X1turbo版の弱点だったスピードも速くなったし，音楽もグレードアップ。しゃべりも入ったし，グラフィックもなんとなく高級感が出ていて，ああ，X68000ユーザーで良かったなあと素直に感じさせてくれる出来なのだ。

（ファンキー浦）

X68000用　5″2HD版2枚組　9,800円（税別）
光栄　☎045(561)6861

スペースローグ

オリジン社の本格的SFRPG「スペースローグ」が発売になった。移植したのはウェーブトレイン，「宝島」でお馴染みのJICC出版局のブランドだ。

主人公は新米パイロット。ワケあって母船を失い，小型偵察艇と我が身ひとつで身を立てなければならない。さて，どうするか。商人として名声を得るもよし，宇宙海賊として名を成すもよし，身のふり方は君に任せられている。あれ？　大航海時代みたいだな。やがて宇宙を旅するうちに自分を取り巻くストーリーが見えてくるという次第。

ゲームはポリゴン処理による3Dフライトシミュレータと，トップビュー型のRPGでなっている。ちょうど「スタークルーザー」の惑星部分を2Dにしたような感じだ。

まだ現物は届いてないけれど，ウルティマを生んだRPGの大御所オリジンの作品だけに，どんなイベントと仕掛けが待っているのかが楽しみだ。　(真面目な浦)

X68000用　5″2HD版　9,800円(税別)
ウェーブトレイン　☎03(3288)1426

レインフォーサー

ザイン・ソフトの今度の新作は「レインフォーサー」。謎のテロリスト組織ニュートを倒すために2人の特殊工作隊員が潜入する，という設定のアクションゲームだ。

ビルのオフィスに始まるステージは，ストーリーに合わせて変わってゆく。8方向にスクロールするマップ上で5種類のウエポンを使いこなし，相手の死角をぬって敵を倒していくのだ。セガの「クラックダウン」などをほうふつとさせる。

私がちょっと試した限りでは，なかなか真面目によく作りこまれていた。各兵器の特徴づけもよく出ているし，操作性や動きに不満点はない。

しかし私は悲しいぞ。「バルーサの復讐」のデモで見せたザイン・ソフトのあのセンス，あれはどこへ行ってしまったのだ。これはこれでいいけど，私はザインというと，こう，サルがほっぺた引っ張っちゃうような，あんなのがいいなあ。いいなったら。　(真面目＆ファンキー浦)

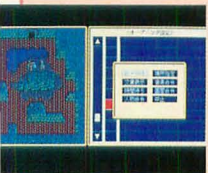
X68000用　5″2HD版4枚組　8,800円(税別)
ザイン・ソフト　☎078(242)2855

ブルトン・レイ シナリオエディタ

「シナリオエディタも出してね」とレビューに書いたのはこの私だったが，さっそくシステムソフトからシナリオエディタが届いてしまいました。見事な攻撃だ。

このシナリオエディタでは，登場人物やマップのパターンは「ブルトン・レイ」か「同シナリオ集」に収められているものを使う。ユーザーはマップ，魔法，アイテム，メッセージ，それから敵と登場人物の設定などをすればよい。何をしたらどんな処理をしてというフェイズ設定を行えば出来上がりという仕組みだ。シナリオにおかしな点がないかどうかはプログラムがきちんとチェックしてくれる。RPGのシナリオエディタとしてはかなり手軽で親切な部類に入るだろう。

単にブルトン・レイのシナリオバリエーションを増やすというだけでなく，ゲーム，特にRPGを作ろう！　などと思ってる人は，シナリオ作成の勉強のつもりでこういった市販のエディタでシナリオを練ってみてはいかがだろうか。1本のストーリーを仕上げるには，いかに細かな配慮が必要かよくわかるし，きちんと動いたときの喜びも味わえるぞ。ちなみに，プレイするのには「ブルトン・レイ」が必要なのでお忘れのないよう。　(真面目な浦)

X68000用　5″2HD版2枚組　4,800円(税別)
システムソフト　☎092(752)5278

アトミック・ロボキッド

UPLのアーケードゲーム「アトミック・ロボキッド」が，システムサコムにより移植された。いわゆる，パワーアップシューティング。マップの中をひたすら進み続け，5種類ある武器を効率よく使い分けながら，現れてくる敵をバッタバッタと薙ぎ倒すというもの。ところどころに分岐点があり，それぞれ難易度が違ってくる。

全部で20面あり，ある一定面数ごとに，画面を余裕でぶちかまし，はみ出るほどのボスキャラとの対決，自機と似たようなロボットとの対決シーンもあり，プレイヤーを飽きさせない。

キャラクターもメカニカルでかわいい。そして，コミカルな自機キャラクターの正面顔がとっても“ぷりてぃー”である。マップのあるところには，残機数と引き替えにアイテムを譲ってくれるゴジラ君がいる。僕は最初出会ったとき，敵と間違えて撃ち殺してしまったが，こいつの泣いてる様子が実はかわいかったりする。こんなゲームに関係ないところでも楽しませてくれる。ファンには期待の1作ですね。(純)

X68000用　5″2HD版2枚組　8,800円(税別)
システムサコム　☎03(3635)7609

ファランクス

「ジェノサイド」「ラグーン」とたて続けにヒット作を出して，X68000ユーザーみんなの期待を一身に背負ってしまったズーム。さて，そのズームの第3弾は，シューティングに決定。タイトルは「ファランクス」。基本的には横スクロールタイプのシューティングだ。

現段階ではオープニングが完成し，ゲームの中身に取り掛かっているところ，とのこと。このオープニングでは，拡大，縮小，回転とX68000の機能をフルに使ってくれている。まだゲーム自体は見ていないのでなんともいえないけど，このぶんでは期待していてもよさそう。

発売は春ぐらいになるらしいから，来月あたりにはゲーム画面をお届けできるんじゃないかな。がんばれ，ズーム。　(香)

X68000用　5″2HD版　価格未定
ズーム　☎011(613)0191

甲子園優勝までの長い道程

Kageyama Hiroaki
影山 裕昭

ここにきて野球ゲームがたくさん発売されているが，このゲームはちと毛色が違う。プロ野球でもなければ，アクションゲームでもない。アートディンクの十八番，箱庭的高校野球シミュレーションゲームなのである。

高校野球＝坊主頭。それが僕のイメージ。実は高校野球を見ることがほとんどない人間なのだ。それなのに「今日の○×対△□の試合はどっちが勝ったの？」とうんざりするほど尋ねられる。高校野球に興味のない人間はめずらしいかのようだ。でも，僕みたいなのに限って，母校が出場したりすると，はりきって甲子園まで応援しに行っちゃうんだよな。

まずは下準備

ゲームは全国高校野球大会地区予選に向けての練習から，地区予選，甲子園での全国大会までをシミュレートしている。最大のウリは全国3990校もの実在の高校のデータをディスクに収めたことだ。その中から好きな高校を選び，チームの監督として甲子園大会優勝を目指すのだ。甲子園で優勝したときには，真に3990校の頂点に立ったということになる。

このゲームでは（初めての試みというわけではないが）自分は選手を直接動かすのでなく，監督として指示を出すのである。監督の仕事はたくさんある。しかし，画面に姿を見せることはない。監督はディスプレイの前にいる自分自身なのだ。

さて，まずはゲームに先立ってユーザーズディスクを最初に作る。生ディスクを自分で1枚用意しよう。ユーザーズディスクは，ユーティリティディスク（製品に含まれている）をドライブ0に入れて立ち上げると作れるようになっている。とっても簡単。画面の指示に従ってドライブ1に生ディスクを入れて，マウスをクリックして待つこと2分半。次に全国49地区から出場地区を選択する。甲子園にいちばん近いのは鳥取だろうか。ここは23校で地区予選を争うので。逆にいちばんの激戦区が神奈川。倍率202倍はかなりきつそうだ。

地区の選択がすむと，その地区から出場する高校名が表示される。高校名をクリックすると，その高校の野球部の部員数，総合評価が表示される。それを参考にして監督になる高校を選択する。高校を決めたら6種類のユニフォームから好きなのをひとつを選んでおしまい。このあと，さらに1，2分待たされて，やっと出来上がりだ。全部で5分はかかるはずなので，コーヒーでも脇に準備して作業を始めるのがいい。ちなみにユーザーズディスクはゲームの途中経過を1カ所しかセーブできないから，ゲームを始める前に何枚も作っておくほうが無難だろう。一度ユーザーズディスクを作ってしまえば，Human68kのDISKCOPYで何枚でも複製できる。

X68000用　5"2HD3枚組　9，500円(税別)
アートディンク　☎0472(79)9392

監督って大変

PL学園とか天理とか甲子園でおなじみの強豪高校の監督もいいだろうけど，やっぱり母校に硬式野球部があるのなら，甲子園で優勝させてやりたいものだ。卒業式でしか歌った覚えのない校歌だけど，甲子園で母校の校歌を流したい。いまになって愛校心に目覚めてしまった僕は，母校を選択した。僕の母校は埼玉県の大宮西高校。このゲームでは“大宮西山”となっているけど，気に入らなければ校名を変更することだってできる。総合評価はCだったがショックはまったくない。事実，うちの野球部は弱かったのだ。こいつらを鍛えて甲子園を戦い抜けるような強靱な精神力と肉体を作っていくのだ。監督として十分やりがいのある仕事じゃないか。

地区大会までの40日間は練習モードだ。この間に選手の実力を把握して，主将，守備位置，レギュラーを決定する。操作はすべてマウスで行い，メニューには，

練習，選手情報，決定変更，システム

がある。監督たるもの，まず選手の長所，短所を知らなくてはならない。マウスカーソルを“選手情報”に持っていき，クリックする。すると，学年別，正選手，内野，外野……（ほかにもいろいろ）といったメニューが出る。内野をクリックすれば，内野手の情報がずらずらっと表示されるわけだ。名前や打率はもちろん，ヒット，2塁打など各安打別の本数も一目瞭然だ。さらに選手1人ひとりが体力，打力，走力，守備力といったパラメータを持っていて，これらから選手の実力を判断する。それができて初めて効率のいい練習メニューを組み立てることができる。

全員の情報をざっと見回すと，さすが3年生は実力がありそうだが，1年生はまだまだ貧弱なのが多い。そういったことを念頭において“練習”をクリックするとポップアップメニューがまたまた表示される。守備位置別だとか個人別だとか，細かい選択ができるようになっている。練習内容は筋力強化，足腰強化といった基礎体力を養うものから，守備練習，打撃練習，投球練習，技術講習といった技術的な面を磨くものがある。40日も期間があるんだから，最

ユニフォームの選択，ポーズが決まってる

初から個人別に細かく練習内容を指示していては気が遠くなってしまうし，とりあえず1年生は足腰強化，2年生は筋力強化，3年生は守備練習と，学年別におおまかに指示していくことにする。

全員に練習内容を指示すると画面には校庭が映し出され，選手たちの練習風景を観察することになる。選手はすごく細かく描き込まれている。しかも，画面写真ではよくわからないかもしれないが，アニメーションしているのだ。素振りをしていたり，投球練習していたり，タイヤを引きずっていたり，ネット越しに女生徒が手を振っていたりもする。画面モードは768×512の高精細モードで16色しかないのだが，選手のアニメパターンも自然で無理がなく校庭のグラフィックもよく描けている。

練習を重ねていくと，選手のパラメータは少しずつ上昇していく。地区予選30日前になったころに，選手情報をもとに個人別に弱点部分を補強していくような練習をさせることにした。選手によっては守備位置を変更したほうがいい場合もあるだろう。それも“決定変更”で簡単に行うことができる。しかし，副産物としてコンバートされた選手の“自信”のパラメータが下がってしまう。いいかげんな気持でコンバートするとあとで泣くはめになる。

まずは練習試合◆◆◆◆◆◆◆◆◆◆◆

要領がわかってきた予選20日前になって，花咲徳英高校から練習試合の申し込みがきた。どうしようか？　選手の実力が計れるし，練習試合はこちらから申し込むことができないことを考えれば，試合をしてみたほうがいいようだ。そうなるとまずベンチに入る15人を選ばなくてはいけない。選手データを参考にして，見込みのあるやつを選んではカチカチと左クリックする。15人選んだら先発選手を決めて打順を組む。

これで終わりではない。1塁，3塁のコーチャーズボックスに立たせる選手も決めるのだ。コーチの走塁技術がランナーの進塁を左右するらしいから，いい加減な気持ち

明けても暮れても練習の毎日

1点取られた直後だ，なんとしても出塁しろ

じゃ選べない。ここまでやってようやく試合開始となる。初めての体験なだけに選手の選択にだいぶ手間取ってしまった。監督の苦労を垣間見た気がする。

試合はリアルタイムで進められていく。両軍ベンチから選手がホームベース前に整列して挨拶から始まる。野手が守備に散るとピッチャーは投球練習を始める。プレー中は相手バッターがフライを上げれば，勝手に動いて捕ってくれる。盗塁だって勝手にやるし，バントだってそうだ。選手は指示がなければ自分の判断で試合を進めていくので，とりあえずほっといても試合は終了するのだ。が，ここいちばんというときは監督が腰を上げねばなるまい。インプレー中でなければ，マウスの左クリックでいつでもタイムをかけることができる。そして“打て”とか“牽制を多くしろ”と指示を与えていくのだ。

やっていると気になる部分はちょこちょことあるが，全体的にはよくできているゲームだ。選手の動くスピードと打球の速さもバランスがとれている。まるで本物の高校野球をヘリコプターから観戦しているような気分に浸れるほどだ。選手は凡フライを落とすマヌケぶりだし，ピッチャーをマウンドに送れば，ぼこぼこに連打を浴びる。きちんとパラメータどおりの働きぶりをする。しかし，豆粒のような選手がひとつの白球を追いかけてプレイしている姿は“馬鹿な奴ほどかわいい”の言葉どおり愛着が湧いてくるのだ。

自分が選手を動かすことができないので，ときにはもどかしい思いをすることもある。実際の監督もさぞかしもどかしい思いをしているんだろう。しかし，9回裏にサヨナラホームランを打ってくれたときなんかはディスプレイの前でガッツポーズをしたりしたっけ。ああ，恥ずかしい。

練習試合が終わったら選手のデータを見てみる。「エラーをした選手は自信が下がっているなあ，もうちょっと守備を練習させなきゃ。あいつは長打力があるからさらに磨きをかけるか」，などと練習メニュー

ついに準決勝，気合いも十分だ

の組み立てがしやすくなるのだ。練習期間はこのようにして過ごしていく。終わってみればあっという間の40日だ。

我が母校はいかに◆◆◆◆◆◆◆◆◆◆

地区予選はトーナメント形式で競われ，本物の高校野球とまったく同じ。試合の進め方は練習試合となんら変わりない。自分のチーム以外の試合は結果だけがわかるようになっている。全部の試合を眺めなければいけないわけではない。

我が母校は地区予選で優勝し，ついにあこがれの甲子園初出場を果たした。甲子園では入場行進と選手宣誓が（ちゃんとアニメーションで）行われ，1回戦の組み合わせが発表される。我が母校は準決勝で敗れたものの，初出場でベスト4に入ることができた。真紅の大優勝旗を手にすることはできなかったが，悔いはない。それが高校野球というものなのだ。優勝すると「全国制覇の軌跡」が見れるというからがんばってきたのに残念だ。その代わり，僕の脳裏にはこれまでの出来事が走馬灯のように浮かんでは消えていく。

後輩が果たせなかった夢をいつか実現させることを誓い，僕の夏は終わった。

注文があります

本文で書かなかったことで，あったらいいなと思ったことをいくつか書き連ねる。データ量的に無理だろうけど，校歌が本物だったら最高。試合中の指示では打者に右狙い，左狙いをさせれるようにしてほしい。甲子園で負けたら砂をかき集めるシーンもほしい。選手の不祥事で出場辞退とかあったら，よりリアルだった？　あと，タッチアップがどうしてもできないのだが……。ないのは絶対におかしい。もし，なんらかの操作でできるのであればマニュアルに明記すべきだ。

文句をつけたあとでいうのもなんだが，このゲームはこのままでも十分に楽しめた。買って損はない。

総合評価　0　5　10

シミュレート	★★★★★★★★★
グラフィック	★★★★★★★★
雰囲気	★★★★★★★★★
感情移入度	★★★★★★★★★★

KLAXはビッグXの夢を見る

またまたパズルゲームの登場、このゲームはアーケードからの移植です。効果音が楽しく、単調になりがちなパズルゲームを盛り上げてくれます。肩ヒジを張らずに楽しみたい、そんなゲームです。

Yamada Junji
山田 純二

「まいど～おさわがせしま～す」とばかりに，またまたアクションパズルゲームが発売された。その名も「KLAX」！　このゲーム，オリジナルは1990年にアタリからアーケードゲームで登場し，日本ではSEGAからアーケードで出されている。そして，今回のX68000版はハドソンからの発売となったわけだ。

ゲームの内容というのは，宣伝文句のとおり「タテ，ヨコ，ナナメにKLAX！」である。え，なんのことかわからないって？　では，もう少し具体的にいうと，同じ色のブロックを縦か横か斜めに3個以上並べるとそのブロックが消えて（これをクラックスと呼ぶ)，ある条件を満たすと面クリアというルールになっているのである。と，これだけ聞くと「な～んだ，COLUMNSと同じじゃないか」と思う人がいるだろうが，基本的にはそのとおりだったりする。

が，こちらはコンベアから転がってくるブロックをパドルで受け止め，倉庫に積み上げていく，というゲーム形態が取られている。しょせんは，どこかで見たようなものではないかという気がしなくもないが（パズルゲームがたくさんあるせいもあるんだろうけど)，のめり込める要素も十分持っているし，出来自体もけっして悪くない。このところの怒濤のシューティング責めで疲れた頭を休めるのには，ちょうどいいゲームだろう。

X68000用　5"2HD版　8,800円(税別)
ハドソン　☎03(3260)4622

さあ,KLAXだ！

さて，いきなり結論めいたことをいってしまったが，もう少しこのゲームについて見ていくことにしよう。

画面構成は，上半分がコンベアである。ここを転がってくるブロックは，全部で10種類あり，これとは別にどんな色にも使えるワイルドブロックなんてものもある。画面中央にはパドルがあり，プレイヤーはまずコンベアで転がってくるブロックを，このパドルを左右に操作して受け取るわけだ。そして，画面下が倉庫。ここにブロックをに落としていけばいい。

倉庫に積み上げられるのは5×5個のブロックで，全部積み上がってしまうとゲームオーバーとなる。パドルにも5個までブロックを積み上げることもできる。複数積み上がったときには，いちばん最後に積み上げたブロックから落とすことができる（図1)。いわゆるスタック構造というヤツだね。

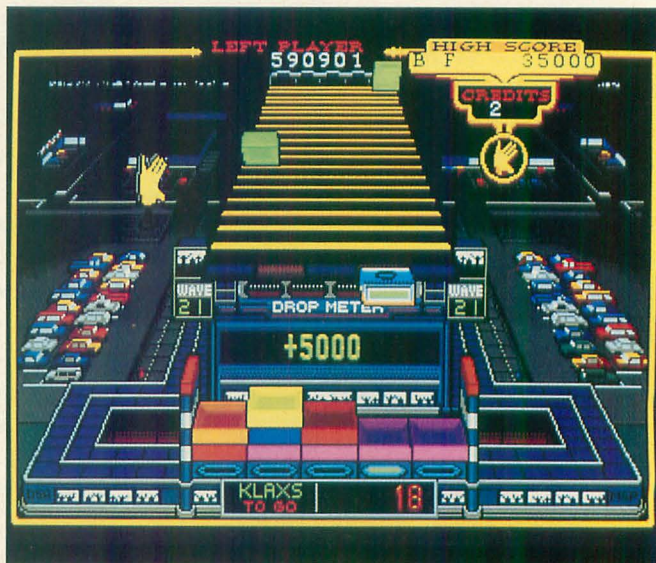
パドルの下には得点表示，画面のいちばん下にはクリアすべき条件，個数が表示されている。で，パドルの少し上にはドロップメーターなるものがある。ようするにパドルで受け損なったブロックの個数が表示されるわけ。5回ブロックを受け損なうとやっぱりゲームオーバーとなる。

操作については，単純明快であまり説明のしようもないのだが，下方向のキーを押すとブロックの転がってくる速度が上がり，上方向のキーを押すとパドルに積み上げたブロックをコンベアの半分まですっ飛ばすことができることだけは，ぜひ覚えておいてほしい。

とはいえ，これを乱用して調子にのって遊んでいると，あとで混乱の絶頂（笑）を迎えることになるので，ほどほどにしておこう。

あそぼ

では，いよいよ遊んでみよう。と，その前に各面（ここではWAVEと呼ぶ）で，クリアすべき条件について説明していきたい。いくら得点を稼いでも，WAVEの最初で提出された課題（ノルマ）をクリアしない限り，次のWAVEに進むことができない。課題のパターンは全部で5種類あり，以下列挙していくと，

1．KLAX TO GO X
　X個のクラックスを作る。
2．HORIZONTALS LEFT X
　ヨコのクラックスをX回作る。
3．DIAGONALS LEFT X
　ナナメのクラックスをX回作る。
4．POINTS TO GO X

これが，KLAXのすべてだ！　が，面ごとにもちろん背景は変わる

X点の得点を獲得する。
5．TILES TO GO X
タイルをパドルでX回受け止める。
となっている。これらのノルマは，5面ごとに繰り返されていて，WAVEが進むごとにだんだんXの数が多くなり，条件が厳しくなっていく。ちなみに，ここでいうクラックスというのは，ブロックを揃えることを指しているのだ。

慣れないうちは，3番目のナナメのクラックスを作るのに苦労するかもしれない。基本的なパターンとしては，図2のようにしていくのがいちばんよいと思う。慣れてきたら，3ブロック以上のクラックスを狙ってみるのもいいだろう。

このナナメクラックスは，あとあとのためにもぜひともマスターしておかなくてはならないだろう。なぜなら，3個のブロックで構成されるクラックスのうちでいちばん得点が高いからである。3個のタテクラックスでは50点しかもらえないが，ナナメにすると100倍の5000点であるので，結構効率的なのだ。

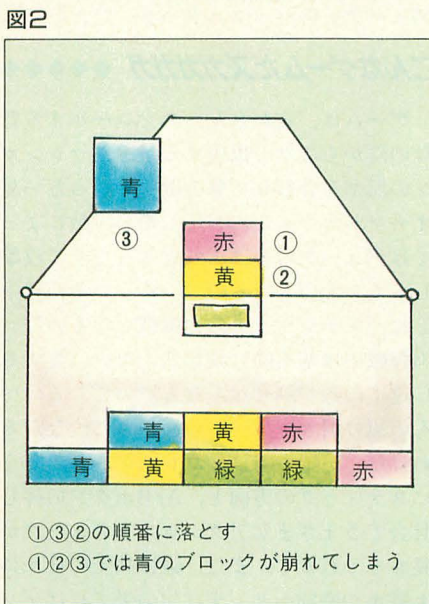
そして，このてのゲームにある連鎖反応も忘れてはならない。次々とブロックが消えていき，得点が跳ね上がる。これがあるからやめられませ〜ん，てなもんだ。目指せ，ビッグX！　得点計算については，図3を見てほしい。

効果音も忘れずに ◆◆◆◆◆◆◆◆◆◆

ゲームの内容もシンプルで奥も深く面白いが，このゲームの効果音もなかなかおちやめである。

まずは，ブロックが転がってくる音だが，それぞれのブロックごとに効果音が違うのである。ぺっぽこ，ぽこぽこ，どどどん，てな具合にとおっても楽しい。BGMがない代わりにこれらの効果音がプレイを盛り上げてくれる。

また，4個のクラックスをやったときには女性の声で「Woo〜」とか，5個のときには「Yaaa〜〜」などと色っぽい声で叫んでくれるし，パドルにブロックがなにもないときにスペースキーをたたくと「ボョョ〜〜ン」などと漫画のような効果音が飛び出してくるのだ。

WAVEクリア時には観客の拍手，ゲームオーバーのときには「Aaa〜」と聞くからに残念そうな声までやってくれる。聞いてて恥ずかしいこともなくはないが，楽しいことには変わりない（しかし，自室でやっていたら親はいったいなにやってんだろ，と思うかもしれないな）。

自分のレベルに合わせて面をスキップできるぞ

面の最初にはノルマとヒントが表示される

まとめ ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

で，ひととおりゲームをプレイしてみた感想としては，操作もルールも単純明快だし，また本文中に何回も述べているように，確かにそれなりの面白さがある。しかし，やっぱり去年パズルゲームが多かったせいなんだろうか，今となってはありきたりの感じを受けてしまうのが残念。

ちょっと気が向いたとき，のほほ〜んとプレイするのがいちばん向いているのかな，こういったタイプのゲームは。

パズルゲームというのは一発アイデア勝負どころだから，もっともっと奇抜なアイデアが出てきてもおかしくはないんだろうし。難しいところだねぇ。ふぅ。

図1

図2

①③②の順番に落とす
①②③では青のブロックが崩れてしまう

図3

私，負けましたわ

KLAX，このゲームについてはあまり語ることはない。すでに，家庭用ゲーム機にも移植されているので知っている人は知っているだろうし。これといって強力なウリはないが，静かな面白さを持っている，そういったゲームだ。

が，そんななかでも，効果音はとんでもなく気に入ってしまった。寝不足ナチュラルハイ状態のプレイ中に，思わず右手を振りかざしてノッてしまうほどだからね。

アタリのゲームには，親切なことにそのままでよいような言葉まで，わざわざ妙な日本語を使ってくれるくせがあるが，このゲームもその特徴は生かされている（笑）。なんでも社長が猛烈な日本ファンだとか。ま，この点については笑って見逃してあげましょう。

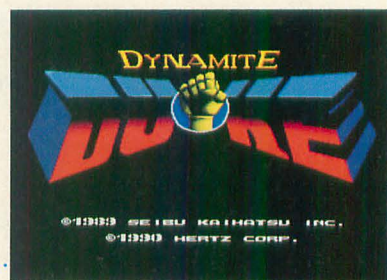
総合評価	0 5 10
グラフィック	★★★★★★★
効果音	★★★★★★★★★★
ゲーム性	★★★★★★★★
ビッグX	★★★★★★★★★
日本語	★★★★★★★

THE SOFTOUCH

●ダイナマイト・デューク

男は筋肉だ,ムキムキだぜ,イエイ!

あのレナムから1年，ようやくヘルツのX68000用ゲーム第2弾が登場。今回は，うって変わってアクションもの，アーケードゲームからの移植作品だ。ジョイスティックでガシガシやりたい。

Nishikawa Zenji
西川 善司

初めにいっておくが，私西川善司はホモではない。が，筋肉ムキムキの男なんかに多少の憧れがある。というのは私は痩せ男で，力もあまりなく，小学校の体育のプールの時間のときにはよく「人体骨格標本」なんて，たいそうな冷やかしを食らったものだ。

だから力持ちの男にはちょっと畏敬の念をもっちゃってたりするわけだ。そんな憧れからかシュワルツネッガーやスタローンの内容のほとんどない映画なんかもよく見るし，格闘もののゲームも結構やったりする（遊園地とかにある腕相撲マシンは間違ってもやらんけどね）。

で，今回発売されたダイナマイト・デュークはそんな筋肉ムキムキの男が単身敵基地へ乗り込んでいく横スクロール型格闘シューティングだ。このゲームはあまり有名でないかもしれないが，ゲームセンターのゲーム，つまりアーケードゲームからの移植なのだ。オリジナルはセイブ開発というメーカーなんだけど知っている人はいるかな。最近ヒットしているシューティング「雷電(RAIDEN)」を作ったメーカーだ。え，「雷電」も知らない？　あの「究極タイガー」のSF版見たいな奴(あ,禁句かな，これって)。まあいいや。移植は「全身凶器の怪力女」という流行語を世に送り出したゲーム「レナム」を制作したヘルツだ。

X68000用　5″2HD版3枚組　8,800円(税別)
ヘルツ　☎03(3371)3012

こんなストーリーだズガガガガ◆◆◆

舞台は近未来。環境の激変に対応すべく新人種が創り出された。これが強化人間と呼ばれる人種で，開発に着手していた軍人のひとりがそいつらを従え，世界征服の計画を企てちゃったのだ。で，ゲームの主人公であり強化人間第1号の「レッド・ダイナマイト」(本名はデューク・フリードリッヒ・フェルゼンだそうだ。道端じゃ絶対フルネームでは呼びたくない奴だな）は，この計画に背いて一度秘密基地から脱出するものの，組織撲滅のため単身舞い戻る……。なんか仮面ライダーの話に似てないこともないが，とにかくこれが「ダイナマイト・デューク」のバックストーリーだ。

こんなゲームだズガガガガ◆◆◆◆◆

ゲームは，右から左へスクロールする背景の陰から次々と出現する敵を，マシンガンの照準を合わせて撃ち倒すといった一見オーソドックスな内容。しかし，敵によってはデュークのそばまで近寄ってきて攻撃してくるものもいるので，そういった連中とはパンチやキックの肉弾戦となるのだ。肉弾戦中は基本的に銃が使えない（要するに遠くの敵が倒せなくなる）ので，このへんの駆け引きがゲームの面白さにつながるわけだ。肉弾戦時はプロレスゲームのようにスティックの方向と，A/Bボタンの押し具合でさまざまなアクションをデュークが見せてくれるぞ。また，敵は兵士のほかにも戦車や戦闘ヘリ，挙げ句の果てにはゾンビや忍者，わけのわからんゲロゲロエイリアンまで飛び出してくるぞ……。

ところで，敵を撃ったり殴ったりするだけならどんな人間でもできなくはない。しかし，主人公デュークはいうのも恥ずかしい強化人間なのだ。強化人間といえば「必殺技」だ。彼の右腕は機械でできているが別に「パックスのしわざ，その1」のせいでもなんでもなく，その必殺技を繰り出すためについているのだ。その名も「ダイナマイト・パンチ」，Bボタンを押し続けるとパワーゲージが上がりだし，ゲージMAXのところで離すと打つことができる。これは画面上の敵を一掃できる必殺技で，まあ，A-JAXや究極タイガーなんかのスーパーボムのようなものといえばわかってもらえるだろう。

ゲーム中，自分に攻撃を仕掛けてこない物がバシバシ壊せるのもこのゲームの面白さだ。ベンツやジープ，電話ボックスに配電ボックス，実生活では絶対に壊せないものまで壊せるぞ。え？　そんなのいつもやっているって。そりゃ犯罪だぞあんた，バビョーン。で，多くの場合こういったものの中からパワーアップアイテムが出現する。弾丸やダイナマイト・パンチの素なんかもあるから「怪しい」と思ったら迷わず撃ってみるといい。

各ステージの最後にはボスがいて，こいつらもデュークと同じ「強化人間」だ。確か「強化人間」は「環境の激変に対応すべく」開発されたというのにどいつもこいつもひょうきんものの個性派揃い……。各ボスは主人公同様「必殺技」を持っていて，それをデュークに対して執拗にしかけてくる(というよりほかに技がないのかもしれない……)。

ステージは全部で9。各ミッション変化に富んだ敵キャラ，背景で，もうアドレナリンじゅくじゅくだぜ，ベイビー。

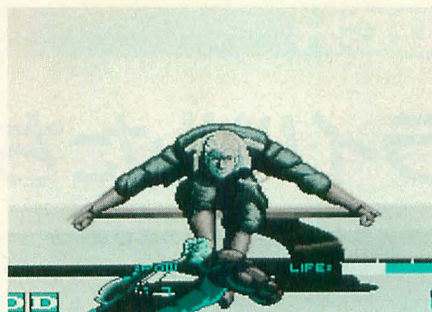
ほうら，背中が透けてる

ステージ攻略だズガガガガ◆◆◆◆◆

ボタンを押すと「CREDIT＝0」と出てくる。さらにBボタンを押すとクレジットが増えていく。9までしか入らないようだ。つまりコンティニュー回数は9回ということ。ステージは全部で9だから妥当な回数といえよう。ゲームを開始するとステージ1の始まりだ。当たり前だ，突然最終面が始まるわきゃない。

1面は敵飛行場の破壊。スクロールが始まるとジープが目につく。これをまず破壊してみよう。防弾チョッキが出てきたはずだ。これをさらに撃ち続けると下に降りてきてアイテムゲージに収まり自分が身に付けたことになる。このようにアイテムは出しただけでは駄目でちゃんと撃って取らなければその効果を得られないのだ。

おっと，さっそくいかにも雑魚という感じの兵士が飛び出てくる。ジョイスティックで照準を合わせてショットしよう。数発弾を当てるとこの雑魚兵士は「**ワン**」といって倒れるぞ。犬か，こいつらは。気づいたと思うがデュークは背中に穴が空いている（「あんた，背中が透けているぜ」なんちゃって）。これは別に病気でも心霊写真でもなんでもない，デュークの目の前に敵が接近してきたとき，その位置をわかりやすくするため便宜上こうなっているだけなのだ。でも目が慣れるまではちょっと奇妙な気分だ。

おぉっと，雑魚キャラものがして溜めすぎるとちょっと大変だぞ。画面中敵弾だらけになっちまう。これまた気づいたと思うが敵の弾は自分の位置に接近してくると色がオレンジから白っぽい黄色へと変わる。この性質を利用して敵の弾の遠近をつかんで効率よくよけよう。そうだ，いい忘れるところだった。敵の弾をよけるのにもってこいの技がある，「伏せ移動」だ。ジョイスティックを斜め下方向に入れると通常より高速に移動ができるのだ。するるるるーってな感じだ。

街中でマシンガンをぶっぱなすアブナいヤツ

セピアに時が止まる！　ってか？

どうでもいいけどヘンなカッコ

けっこう強いヘリコプター。でも本ボスじゃない

しばらくするとスクロールが止まり，中ボス「武装ヘリコプター」が登場してくる。地道にマシンガンもいいが，せっかくだからいっちょ派手に「ダイナマイト・パンチ」を使ってみよう。……さすが必殺技，一撃でヘリコプター爆発だ。ズバーン!!!

そのあと少しの間，雑魚との戦闘が続く。で，本ボス登場。こいつは「スタンガン」とかいう名前のデュークと同様強化人間だ。こいつは右腕がロケットパンチ，左腕がマジックハンドという，もはやまともな日常生活は送れない，人生を投げたような奴だ。このゲームを始めたばかりの人は，恐らくこいつにかなり苦しめられることだろう。

スタンガンはマジックハンドを振り回す前に腕を水平に構える。これを確認したらすぐ「伏せ」の体勢を取ろう。ノーダメージでよけられるぞ。どの面の本ボスも自分の「必殺技」を使うとき，必ず独特のポーズをとるようだ。頭に叩き込んでおこう。ジャンピングキックやダイナマイト・パンチがうまくスタンガンに命中すると，奴は遠くへ放り出される。このときスタンガンは起き上がりざまにロケットパンチを撃ってくるので注意しよう。これも「伏せ」で避けられる。

と，このように，本ボスは闇雲にジョイスティックをガチャガチャやって戦わず，しっかり自分なりのパターンを組み立てて戦えばそれほど難しくないはずだよ。

では最後に全ステージクリアのためにポイントを示しておこう。

- ステージ3の橋の上の砲台はダイナマイト・パンチでやっつけよう。ダイナマイト・パンチが砲台に隠されているので結局±0だ。
- ステージ4。中ボス戦車の出てくるシーンで手前の柱に防弾チョッキが……。
- ステージ5は最初のカベとなるだろう。中ボスヘリコプターのシーンで「FULL AUTO」のアイテムがあるが，これは取らないこと。次に出てくる本ボスは結構堅いのでダイナマイトパンチを使って倒そう。「FULL AUTO」を取ってしまっているとダイナマイトパンチが打てないぞ。
- ステージ7の赤忍者の手裏剣は，相手が投げのポーズのときに一度伏せ移動で左に行ってから右へ行けば簡単によけられる。
- ステージ8のゾンビシーンに出てくる救急箱は「毒」。取ると体力が減っちゃう。
- 最終面のボスは2回変身するので体力配分，弾数配分を慎重に。1回目の変身時はジャンプキックの連続をお見舞いすれば楽勝。2回目変身時に繰り出してくる触手は正面に来ても「伏せ」てればやられない。
- アッパーとジャンピングキックは必ずマスターすること。対本ボス戦の有効な戦力だからね。
- 溜まった敵をマシンガンで撃つときは照準を左右上下に振りながらだと効果的。

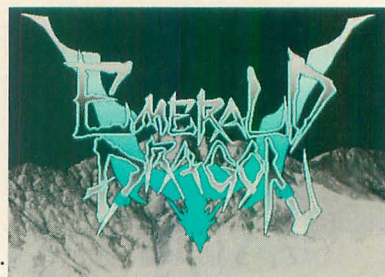
マイナーアーケードゲームの移植大賛成

見た目が派手なだけの内容のない新作に，テーブルを譲った数多くの名作ゲームたち……。こういったマイナーアーケードゲームのX68000への移植は（まぁいろいろ意見もあるだろうが）私は**大賛成**だ。お気に入りのゲームだったのに，ゲームセンターから突然姿を消してしまったゲームが沢山あるのだ，私には。ヘルツさん……，次は「ぶたさん」「ワンダーモモ」「ラビオレプス」「ダーウィン」あたりを移植してくださいよん，ゴロニャん，うふん。

総合評価	0　5　10
ゲーム性	★★★★★★★
操作性	★★★★★★★
サウンド	★★★★★
グラフィック	★★★★★★★
難易度	★★★★★★
ムキムキ度	★★★★★★★★★★

戦え! 永遠のライバルたち

Komura Satoshi
古村　聡

PC-8801版で人気を博したゲームの移植版。が，もちろんこのゲームのウリであるビジュアルやキャラクターグラフィックはX68000用に新たに描き起こしたもの。ジョイスティックやマウスにも対応している。

「ゆ，許さんぞ！　オストラコン！」
「ほう，許さなければどうする？　俺は以前とは違うぞ，かかってきてみろ！」

くぅーっ，燃えるぜ。ついに出た出たやっと出た，エメラルドドラゴン，通称エメドラのX68000版がこのたびグローディアから発売になったのです。思えばPC-8801版が発売になってから1年余り。長い年月でありました。しかしながら，待った甲斐あってしっかりとグレードアップして登場。うるうる，生きててよかった……。

えー，このエメドラは，いわゆるRPGといわれるやつなのですね。しかし，しかしだ。こいつをそんじょそこらのRPGだと思っちゃいけない。主人公のアトルシャン(こいつが人間じゃなくてドラゴンなのだ，実は)，ヒロインのタムリン，そしてその仲間たちの友情あり涙あり，アニメーションあり，美形のライバルキャラありと少年マンガもビックリのあああ，浪速節RPGなのだ。

さあ，とくとご覧あれ（ベンベン）。

大丈夫，君を守ってみせるよ◆◆◆◆◆

いま，アトルシャンの目の前にタムリンがいる。温かいティーカップを口に運びながら，彼は彼女のことを，そして昔一緒に過ごした日々を懐かしく感じていた……。

彼女は幼いときアトルシャンのいるドラゴン小国にやってきた。ドラゴン小国に辿り着いた難破船の唯一の生き残りとして，だ。そしてドラゴンであるアトルシャンと一緒に，人間のタムリンは育てられたのだ。が，やがてタムリンはイシュ・バーンに帰っていく。タムリンを心配したアトルシャンは，彼女に自分の角を折って角笛として渡したのだった。なにかあったらこれを吹いて，いつでも君を助けにいくよと……。

それから3年の後。アトルシャンはタムリンの吹く角笛の音を聴いた。だから彼は彼女を守るために遥かなる次元の向こう側，そしてドラゴンにとっては呪われた地であるイシュ・バーンへとやってきたのだ。

タムリンの口がゆっくりと開く。
「どこからかこのイシュ・バーンへ魔将軍オストラコンの率いる魔軍がやってきたの。いまやイシュ・バーンは落城の危機にさらされて……。明日にもここに侵攻してくるかもとみんな恐れおののいているわ」
「……なるほど。タムリンとしてはほってはおけない，というわけか。しかし女の身で戦うとなれば相当の覚悟が必要になるぞ。悪くすれば死ぬかもしれない」

わざと表情を変えないように，そしてタムリンの顔を見ないように彼は続けた。
「……魔族に殺された連中は死んでもなお生きるんだ。死霊ってやつさ。俺はやだね」

タムリンと目を合わせる。タムリンはアトルシャンをはたこうとしたのか手を上に挙げていた。そして大きく息を吸って手を下ろし，いかにも落ち着いているふうにいった。が，息がふるえているのがわかる。
「いいわ。もうあなたには頼まないわ。私ひとりでも戦ってみせる！」

予想どおりの答えにアトルシャンはなんだかちょっとおかしいような気さえした。
「悪かったよ，タムリン。君の決意のほどを試させてもらっただけさ。もっとも君がそう答えるのは初めからわかってたけどね」
「……それじゃあ……」

こうして彼らの危険な旅が始まったのだった。魔軍を倒すカギはただひとつ，アトルシャンが白龍から聞いたエメラルドドラゴンの謎，だけだ。でも大丈夫だよ，タムリン。俺は君を守ってみせる，必ず，ね。心の中でアトルシャンはそうつぶやいた。

システムはストーリーと共に◆◆◆◆

このゲームはRPGですが，最近流行りのARPGと違ってマップ上を敵キャラがチョコマカ歩き回っているわけではありません。自分が歩いていて敵と出会うと画面が切り替わって戦闘画面になるのです。

戦闘は敵キャラとマイキャラのパーティの俯瞰図になり，チョコマカと2頭身のキャラたちが動き回って戦闘してくれます。動かせるのはマイキャラのみ。あとのキャラたちは自分の意志(?)であっちの敵を斬りにいったり疲れて戻ってきたりします。また，レベルに自信があれば自動戦闘も可。それでもどこぞの“ガンガンいこうぜ”とかやるAI戦闘なんぞよりよっぽど賢い。

X68000用　5″2HD版6枚組　9，800円(税別)
グローディア　☎03(3220)5226

戦闘シーン。ザコでもなんでもぼてくりこかせ！

これが街の中。よっ！　市民A，元気かい？

おおっと，レベルの話。このゲーム，基本的にアトルシャン，タムリンも含め5人のパーティで行動するのですが，実はそのなかで戦闘を行ってレベルが上がっていくのはアトルシャンとタムリンだけなのです。では残りの3人のヒットポイントの最大値やストレングスは？　そう，固定されたままなのです。じゃあどんどんレベルの上がっていく敵にどう対処しているのかというと，シナリオが進むにつれてドラマが展開されながらメンバーが入れ替わり，旅を続けていくという方式が取られているのです。要はメンバーがどんどんチェンジしていくわけなんですねー。いやぁ，すごい。

宿敵オストラコン。うーん，美形だなぁ……

ああ，水晶に捕らえられたハスラム。いい構図ね

永遠のライバルたち！ ◆◆◆◆◆◆◆◆

アトルシャンたちが初めて敵の魔将軍オストラコンと出会ったのは魔軍前線司令部だった。オストラコンは天才的策略家，しかも冷酷非情な男である。

「オストラコン！　覚悟っ!!」

アトルシャンはそう叫ぶとオストラコンが身構えるより早く斬りかかり彼の右腕を切り落とした。

「ぐごぁわ！　俺様の右腕をよくも！」

しかし，彼は部下の助けで命からがら逃げ出していたのである。

しばらくの時が過ぎた。元フラワルド伯爵邸にオストラコンがいると聞いて急襲をかけたアトルシャンたち。

「オストラコン!!」

アトルシャンが剣に手をかけようとしたその瞬間2匹の魔物が彼らの前を遮った。

「ココヲトオスワケニハイカン！」

オストラコンは薄笑いを浮かべてこちらを見ている。はっ，とタムリンが息を飲んだ。なにかに気づいたらしい。

オストラコンの右腕だ！

切り落としたはずの右腕，それは明らかにこの世のものとは思えぬものだった。魔性……。オストラコンはアトルシャンたちに復讐するために魔に魂を売り渡したのだ。

「待っていたぞ……」

オストラコンが静かにいった。

「オストラコン！」

「おっと。その前にこいつを見るんだな」

オストラコンが異様な右手を突き出す。その手のひらには不気味な黒水晶が……。

「ハスラム！」

ファルナが叫んだ。そう，エルバードの危機を聞いてパーティを去っていったハスラム王子だ！

ハスラムが黒水晶の中に閉じ込められているのだ！

「見てのとおりお前らの大切な仲間は俺様が預からせてもらった……。この小僧が助かるも助からぬもお前ら次第だ。」

「なにが望みだ，オストラコン！」

「お前たちの命……」

少年マンガ道つっぱしるRPGなのだ！ ◆◆

という感じで，ドラマはアトルシャンと魔将軍オストラコンの戦いを中心に描かれていきます。そして，このストーリー展開こそがグラフィックと並んでプレイヤーを前へ前へと進ませる吸引力を作っているわけです。かわいいヒロイン，それを守る主人公，強く冷酷な美形敵キャラとくれば，普段から少年マンガにどっぷりの方には必須のゲームですね。なんてったって最近の少年マンガは“絵がきれい”で“ヒロインが超かあいい”というのが必須条件ですから(たとえば“ああっ女神さまっ!”とか“3×3 EYES”とか……，でしょ？)。

このゲームの最大の特長は，やっぱりグラフィック。これがなんてったってディスク6枚組のうちオープニング，エンディング，ビジュアルA/Bの4枚がグラフィックデータだという力の入れようなのです。もう店頭デモなどでご覧になっているでしょうが，グラフィックは非常にきれいです。それにヒロインのタムリンも見目麗しく愛らしくかあいいかあいい本当にかあいいのだ！　ぐふふふふ，じゅるっ(こらこらっ)。

ご存じのようにこのゲーム，PC-8801版からの移植なのですが，グラフィックもそのままもってくるような横着はしないで，ちゃーんとX68000用に描き直してくれているのでした。ちょっとPC-8801版と絵の感じが変わったところも多いですが，決してグレードダウンということはありません。ちょっとオープニングは顔が丸っこくて，ゲーム中のグラフィックは縦に長くなっちゃってるなー，という気もしますけど。

とにかくヒロインがかあいい，美形の敵キャラがいる，そしてアニメーションと最近の流行りをばっちり押さえた少年マンガ道をつっぱしってるゲームなのです。

なにはともあれ，このゲーム，実は結構時間のかかるゲームなのですが（まあ，あれだけのストーリー展開がありますからね。しかたないっちゃしかたないでしょう)，ひとつ間違えばただマップがだだっぴろいだけの退屈なゲームになるところを，うまく見せ場を盛り込み時間がたつのを忘れさせてくれます。移植も完璧ですし期待を裏切らないという意味ではまったく感謝したいくらいの出来でしたね。休みなんかにどうですか？　絶対おすすめ，イチオシのゲームですよん。

ぐいぐい引っ張ってくれるのだが……

遊んでもらえるゲームを，ということをよく考えて作られたゲームだと思う。アニメシーンをつけることで「早く次のグラフィックが見たい」と思わせる。相談コマンドで「次はどこどこへなになにしに行くべきだ」とほかのキャラクターが教えてくれる。これで次の目標をプレイヤーに教える。エンディングへ向けての引力があるといってもいい。スタッフの頑張っている姿が目に浮かぶようだ。

ただひとつ難をいえば，少々ゲームの作りが荒っぽいように見えてしまうことだ。たとえば戦闘中のメッセージで「残念，逃げられない！。」と“！”のあとに“。”があったり，ハスラムがファルナにいうセリフなのだが「顔色がうかない」などという不思議な日本語があったりするのだ(これは「顔色がよくない」「うかない顔をして」のどちらかが正解だろう)。ささいなことだし，作った人たちは決して雑に作っているつもりはない。いや，できるだけ真剣に作っているはずだ。でなければこんなに売れるゲームに作れるわけがないのだが，そのように見えてしまうのも事実なのだ。

これだけ売れてしまうといろいろとプレッシャーもかかり大変だろうが，次回作でぜひこのようなことにも気をつけて頑張ってほしい。

総合評価（0　5　10）

項目	評価
お手軽度	★★★★★
ストーリー展開	★★★★★★
自動戦闘モード	★★★★★
ミュージック	★★★
タムリン	★★★★★★★★★★
おすすめ度	★★★★★★★★★

テーブルトークの本家,いよいよ登場

「月海」の北側の地には「フラン」という数千年前に最盛期を迎えた都市が存在していた。「新市街」の「評議会」を中心とする人間の軍と「旧市街」の「邪悪」な軍が対立するなか，新生「フラン」を再興するのだ。

Kameda Masahiko
亀田 雅彦

久しぶりのレビューで，いま猛烈に緊張している。いつもの「ですます調」文体まで忘れちまった。ふっふっふっ，どんなゲームでも相手になってやる。

今回の相手はAD&D® (Advanced Dungeons&Dragons) から「POOL OF RADIANCE」だ。これは，アメリカで大ヒットした（と，ものの本に書いてある）AD&D®をパソコンでパクった（?）ゲーム。みんなでワイワイやるテーブルトークを，パソコンを使ってひとりで遊べるようにしたってわけ。なにしろAD&D®といえば，日米共にその筋で知らぬものなし。面白さは保証つき。あとはいかにコンピュータ化したか？　ってことだけが気がかりだね。そこで今回のレビューは，ガンガン！　本音で勝負しようと思っている。辛辣な表現も出てくるかもしれないがそこはご容赦願いたい。

さて，まずはシナリオの面白さを知っていただこう。いまあなたはパーティを組んで，ある使命を遂行しようとしている。題して，

墓地に闇夜が迫るとき，アンデッドが甦る
〜失われた英雄伝説〜

このヴァルヒンゲル墓地の掃討を依頼されたとき，パーティの中には反対するものもいた。魔術師のサマンサはアンデッドを恐れ，盗賊のドロキチは「墓荒らしは趣味じゃない」と言った。しかし，僧侶のボーズは退散呪文にいささかの自信があり，「アンデッドごとき敵ではない」と豪語した。戦士KAME，ドーイもその言葉に勇気づけられて，この話を受諾したのだった。

X1turbo用　5"2D5枚組　9,800円(税別)
ポニーキャニオン　☎03(3221)3161

パーティはこの時点で５人。歴戦のなかで，３人が帰らぬ人となっていた。我々は町の訓練場で２人の傭兵を雇うことにした。彼らの名前はヒーロー兄弟。傭兵の中でも，最も強い兄弟だ。我々の戦士らと同等のレベルはある。かなりの年のはずだが，力強さは若者に負けず，かなりの修羅場をかいくぐってきたはずだ。めったに表情を変えず，無駄なおしゃべりもしないので，それ以上のことはわからなかった。

隠された聖騎士の謎 ◆◆◆◆◆◆◆◆

墓地にはさまざまなアンデッドモンスターがいる。アンデッドモンスターはより上級なモンスター（スペクター）によって召還されている。スペクターをやっとの思いで倒すと，スケルトンやゾンビなどの下級モンスターは召還されず，もはや墓地にはいないようだった。

そんなとき，パーティは十字架の形をした大理石の納骨堂を見つけた。そこには……はるか昔の聖騎士伝説が累々と記されているではないか！　我々は慌ててその巻物を読んだ。

「無敵の名をほしいままにしていた戦士がアンデッド退治のため墓地へ入った。しかし，彼はついに帰らぬ人となった。

彼の弟もまた聖騎士であった。兄の仇とばかりに，女祭司長の力を借り墓地へと乗り込んだ。何千年も生きてきたバンパイアの魔力は想像を絶するほどだ。しかし，彼らの聖剣，魔法を込めたメイスは，ついにバンパイアを仕留めた。その断末魔の叫びは墓地全体に轟いたそうだ」

パーティは読み終わった巻物を静かに置くと，あとずさりするように建物を出た。記述にあるバンパイアに恐怖を抱いたのかもしれない。

最後の死闘 ◆◆◆◆◆◆◆◆◆◆◆◆◆

が，しかし，その恐怖はすぐに現実のものとなった!!　我々が建物を出た瞬間，空間に巨大な魔力が集中する。身構える間もなく，集中した魔力は青白い顔の男へと姿を変えていく。

バンパイアのわきには２匹のウルフがいる。こいつらも魔法を使うようだ。

我々は皆，死を覚悟した。と同時に，興奮する自分を感ぜずにはいられなかった。

ボーズが退散呪文を放ったがまったく効かない。レベルが足りないのか？

バンパイアの右手が上がった。ウルフはファイアーボール（火球爆発）の呪文を投げつける。火炎は僧侶，盗賊，戦士を飲み込んだ。３人はとてつもないダメージを受け，ほとんど即死した。

残った戦士３人はバンパイアに攻撃を集中するため，敵の真正面に立った。バンパイアの左手が青白く輝く。その手に触れられた戦士は急速に痩せ細った。そして，ついにその存在自体が消滅してしまった。生命力を奪われたのだ。もうひとりもウルフと相打ちに。パーティの残りは２人。敵はバンパイアとウルフ１匹。

サマンサは以前の冒険で，ワンド・オブ・ライトニングボルトを手に入れていた。それはその強力さゆえに，一生使わないと心に誓ったものだった。

ワンドを手にすると，白い稲妻が衝撃波

この番号をマニュアルで参照してと，フムフム

となって空間を切り裂き，バンパイアの体に衝突した！　そしてその体を突き抜け，ウルフを壁に叩きつけた。

しかし，バンパイアは生きていた。一瞬のスキをついてマジックミサイルを放ってきたのだ。なすすべもなく戦士は倒れた。

さらに，ライトニングボルトとマジックミサイルが同時に放たれた。墓地の闇は真っ白に染まり，ものすごい量の魔力が解放された。そして，そのあとには……。

＊　＊　＊

私は呟いた。「あれ，どこまでセーブしといたっけ？」

ゲームシステム◆◆◆◆◆◆◆◆◆◆◆◆◆

いままでの話はゲーム中にいくつかあるクエストのひとつ。しかも，かなり簡略化，脚色してある。でも，ゲームバランスのよさ（難しすぎず，簡単すぎず）や，シナリオのドラマ性は伝わっただろう（？）。

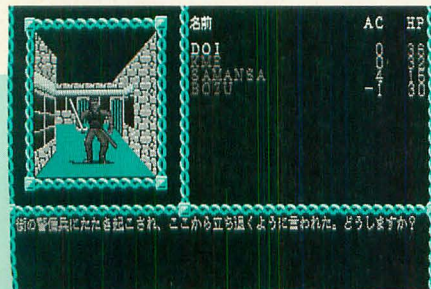
さて，それではこのゲーム本体の説明をしよう。あなたはまず8人以内のパーティを組む。そしてフランという町の評議会に行き，仕事（たとえば，「墓地を掃討せよ」など）をもらう。その仕事自体がクエストになっていて，解決すれば報酬がもらえるし，冒険中にも宝を発見できる。またそうした仕事をこなしていくうちに，いろいろな手掛かりを見つけてフランの謎を徐々に解いていく。

このゲームの特徴や面白い点を列挙してみよう。

●シナリオ，およびイベントのバランスがとれている

ゲーム全体の謎解きと個々のクエストの関わりがはっきりしているので，ゲームの進行に無理がなく，飽きさせない。

●細かいメッセージなどに気配りがある

キャラクターをパーティから外すとき，「安堵の溜息をついた」「別れを告げた」「放り出した」など状況によってメッセージが違う。パーティの行動制約も少なく，少しでもテーブルトークに近づけようとしている努力を認めたい。

せっかくいい気持ちで寝ていたのに

墓地の中は寒いので，あたりは銀世界？

こいつが墓地の親玉，バンパイア

このゲームで忘れちゃいけない存在は，NPC（ノン・プレイヤー・キャラクター）だ。先の例では「傭兵」と書いたが，NPCはまさにそんな感じだ。冒険途中でパーティに入る奴もいる。強い奴，弱い奴，また，敵なんかもいて，NPCの存在はこのゲームの差別化にひと役かっている。また戦闘時の戦略にも欠かせない（なお，NPCの行動はコンピュータが行う）。

●かなり洗練されたゲームである

すべての操作がテンキー（ジョイスティックでも可）オンリーのメニューセレクト方式。ディスクアクセスを最小限に抑えようとする努力も見受けられる。かなりのテストプレイを積んだとみた！

●種類が豊富

キャラクターには人生観まで設定できるし，顔，体，戦闘時のアイコンまで変更可能（男の顔に女の体はぶっきー）。武器もかなりあるし，持ち物の重量や貨幣といった本家の細かい設定を忠実に再現している。それが面倒臭さにつながってないことが，感情移入にプラスに働いている。

とりあえず印象に残ったものを書いてみたが，結局「AD&D®」と「プログラムの気配り」の2つが特徴だと思う。

続いて「これはアカん」と思う点。

●ディスクアクセス

戦闘が始まるときにガーガー，終わるときにもガーガー。アクセスの多さはしかたないとしても，そのスピードが遅い。おまけにディスク5枚組(＋ユーザーディスク)。2HDディスクも入れるとか，バンクメモリ，EMMを使うとかの気配りがほしかった（いいゲームだけに残念）。

●プロテクト

翻訳ディスクなるマニュアルプロテクトがあり，起動のたびにパスワードを入力する。ゲーム中にもこのパスワードを使う場所がある。面倒臭さが増しているだけで，正規ユーザーにメリットがない。

ほかにゲーム中の重要なメッセージが別マニュアルになっている（これは単なる容量の問題でプロテクトじゃないと思うけど）。つまり，ゲーム中に「何番の文章を見ろ」というメッセージが出る。プレイヤーは別マニュアルの何番を見て，その状況を把握する。これは「画面のメッセージを書き写さずにすむ」などの理由で便利だ。これはほかのゲームでも参考にしてほしい。

●フラン市街の外に出るとやる気を失う

「クエストの目標がはっきりしない」，「やたらとモンスターに出会うので，前に進めない」という理由で外のクエストはとてもやる気になれない。ただし，フラン市街だけでも普通のRPG1本分は楽しめる。

やっぱりディスクアクセスが問題かな。

これらをプラス，マイナスした結果，私はプラスの評価をしたい。X1turboでここまでの仕上がりは，注目に値する。特に，プログラム技術をウリにすることなく，あくまで「楽しめる」という視点からゲームを作る姿勢がいい。この「ゲーム性を重視したゲーム」精神を大切にしたい。

テーブルトークは面白い

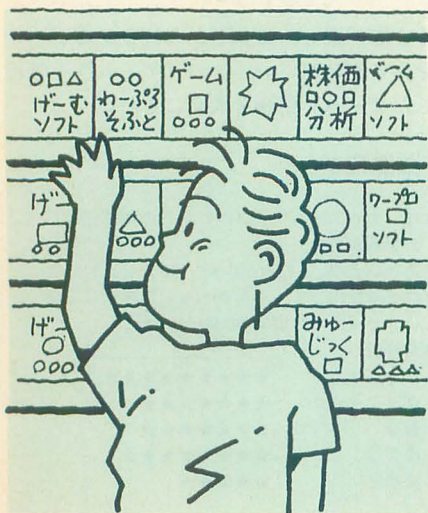
「X1のゲーム」というだけでめずらしい昨今。出るだけでもありがたく，ゲームの善し悪しを判断するのも難しいが，8ビットパソコン上のゲームとしては，かなりよい出来である。16ビット以上も含めたRPGゲーム一般から見た場合は，普通のレベルだと思う。

しかし，ここで注目したいのは，最近のRPGの動向だ。近頃，特に“ドラクエ”タイプの小さいキャラがチョコチョコ走り回るものが多い。正直言って，あのテのRPGは私は感情移入できないのでパス。それとは正反対をいくこのゲームは，大人向け（オタク向け？）の本格派としての価値がある。

あとひとつ。X1turbo用なら400ラインディスプレイ専用にして，漢字をちゃんと表示してほしい。せっかくの漢字VRAMが泣くぜ。

総合評価	0　5　10
シナリオ	★★★★★★★★★
グラフィック	★★★★★★★
音楽	★★★★★★★
操作性	★★★★★★★★
快適度	★★★★★

AFTER REVIEW

X68000オンリーのソフトハウス，ズームは，前作のジェノサイドの出来がよかったこともあり，かなり注目されているようです。さて，アクションRPGとなった第2作「ラグーン」の反響はどうでしょうか？

ラグーン

▶キャラクターがでかいだけあって，迫力がある。マティアスVSゼラーなんてすごすぎて感動した。が，最後の敵をもう少し強くしてほしかった。それに，ムーンブレイドなしで倒せるのはなんとかして！

笹倉　英昭(15)兵庫県

▶ストレスを感じさせない，実にスムーズな進行。ここにズームの14カ月のこだわりを感じる。また，例によってのディスクエンベロープの心づかい。涙。

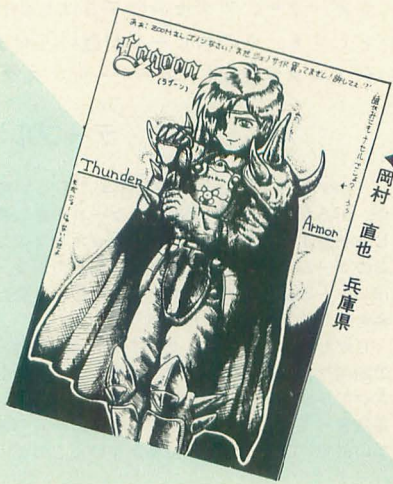
西本　英樹(18)北海道

▶絵，音，動き，ともによくできていると思う。これで4方向じゃなくて，8方向だったらもっとよかった。敵は弱いけど終われないゲームよりはよっぽどいい。

安永　吉徳(21)長野県

▶「X68000オリジナルゲームはムズイ」というのを打ち破ってくれたもんね。けど，簡単すぎたかも……。計10時間ほどでクリアできたもんね。澤田　裕史(15)神奈川県

▶ホホホ，やっとラグーンが出た。エンベロープの“ネコ”が楽しみで買ったのだが，X68000にしてはめずらしく最後のボスが弱い。また，もっと「X68000ならでは」というのがほしかったが，魔法がハデでいい。エンディングは最高だ。あと，なんかイースIIににてるぅ〜。宮前　龍也(15)東京都

▶9月15日の午前に買ってきて，午後1時からやりはじめた。そして8時間後，エンディングを迎えていた。いろいろ演出があり，まあまあ遊べた。しかし，ムーンブレイドを取らずにソア（最後の敵）を倒してしまい，唖然とした。石川　洋(16)愛知県

▶凄まじいまでのグラフィックと役に立たない金。いまいちの難易度。やたら派手な魔法といい，行方不明のライ麦パンといい，なかなか面白い（5番目のリングはどこにある？）。　井上　和也(21)福岡県

▶キャラクターが大きいのはいい。しかし，なんのための剣の振りかわからない。もう少しズームにはがんばってほしい。そう思いませんか。それにジャンプもひどい。

米倉　正人(23)長野県

▶先日，待ちに待った「ラグーン」をやった。感想はひと言，「あっけない……」である。前作の「ジェノサイド」が“超”難しかったため，このギャップの大きさにびっくりしている。普通にプレイしていれば1日で終わることも可能だ。もっと長く遊べることを期待していたのですが……。グラフィックや音楽はきれいでしたが，あまりにも簡単すぎたなあ。

山本　昭治(22)神奈川県

▶ビジュアルシーンなどは圧巻で，プレイには気合いが必要とされないところがいい。い。とにかく音楽が気に入った。

羽田　茂樹(20)愛知県

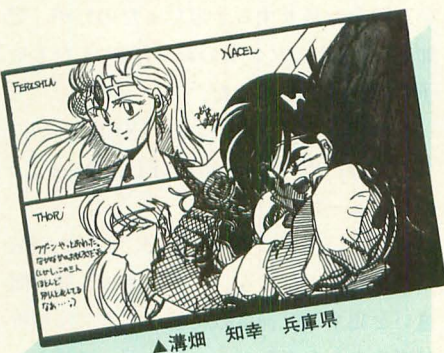
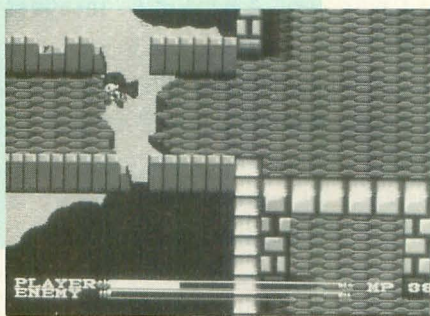
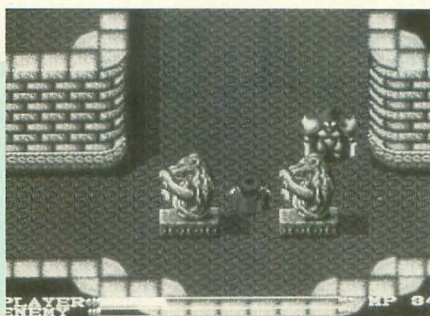
◀岡村　直也　兵庫県

力の入ったオープニングシーン

最初のほうのシーン，洞窟のそばで人が……

▶魔法は派手だが，見せるだけでたいして意味がなくてはがゆい。

鈴木　克史(17)静岡県

▶よくできたゲームでもたいていなかだるみのようなものがある。ラグーンの場合の問題はそのなかだるみがいきなり前半にきちゃうことね。後半の盛り上がり方がすごいと聞いてもう一度続きをなってみる気になったけど，最初は2匹目のボスの前ぐらいで面倒臭くなってほかのゲームに浮気しちゃった。だって，敵（ざこ）は弱いくせに無意味にカタイし，進行が単調。ついおざなりにプレイすると，死んじゃうし。もちろんグラフィックがきれいだったし，操作性も抜群だったけど（多重スクロールの通路はおっこちそうでドキッ)。きっとみんなもズームだからと信じて先へ進んだんじ

▲溝畑　知幸　兵庫県

落っこちそうでこわい

ゃなあい？　結果的には話題を集めるだけのことはあったと思うんだけど，次は最初からハラハラドキドキの展開が続くゲームがやりたいよお。ズームなら！　信じられる!?

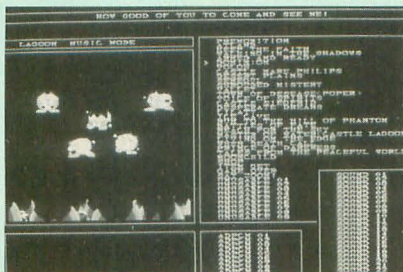
七瀬由美子(19)神奈川県

ラグーンの場合，やはりグラフィック，音楽，そして，オープニングデモやエンディングがいいという意見が多かったようです。しかし，肝心のゲーム内容に関しては賛否両論。演出もすばらしい，という意見もあれば，問題あり，という意見もありました。まあ，やっぱりいちばん多かったのは，イースなどに似ているという意見かな？　しかし，ほめているにしても，けなしているにしても，皆さんのズームに対する期待はあいかわらず大きいようでした。

X68000用　5″2HD版4枚組　8,800円(税別)
ズーム　☎011(613)0191

ミュージックモード発見

長野県にお住まいの松沢宏治さんほか数名の方からラグーンのミュージックモードへの入り方についてのお便りをいただきましたので紹介します。

まず，普通に立ち上げるとデモが始まり，そのあとユーザーディスクあるいはシステムディスクを要求してきますよね。そこで，データディスク3を3回以上，2を2回以上入れます。そして，そのあとシステムディスクを入れるとOKというわけです。

発売中のソフト

★ワールドスタジアム　SPS
X68000用　5″2HD版2枚組　8,800円(税別)
★アトミック・ロボキッド　システムサコム
X68000用　5″2HD版2枚組　8,800円(税別)
★ラプラスの魔
エム・エー・シー　ハミングバード
X68000用　5″2HD版3枚組　8,700円(税別)
★Magical Shot　M.N.M Software
X68000用　5″2HD版　7,800円(税別)
★ザーク レジェンド スペシャル　マキシマ
X68000用　5″2HD版4枚組　8,800円(税別)
★スライス　M.N.M Software
X68000用　5″2HD版　7,800円(税別)
★大航海時代　光栄
X68000用　5″2HD版2枚組　9,800円(税別)
★KLAX　ハドソン
X68000用　5″2HD版　7,700円(税別)
★プール・オブ・レイディアンス
ポニーキャニオン
X1turbo用　5″2D版5枚組　9,800円(税別)

新作情報

★D～欧州蜃気楼～　ウルフ・チーム
X68000用　5″2HD版3枚組　12,800円(税別)
★RYU～哭きの竜より～　ウルフ・チーム
X68000用　5″2HD版　11,600円(税別)
★DRAKKHEN　EPIC/SONY RECORDS
X68000用　5″2HD版　価格未定
★アルガーナ　M.N.M Software
X68000用　5″2HD版　6,800円(税別)
★Misty Vol.7　データウエスト
X1turbo用　5″2D版　5,000円(税別)
X68000用　5″2HD版　5,000円(税別)
★マーブルマッドネス　ホームデータ
X68000用　5″2HD版　価格未定
★ファンタジーⅣ　スタークラフト
X68000用　5″2HD版　9,800円(税別)
★中華大仙　シャープ
X68000用　5″2HD版　価格未定
★エイリアンシンドローム　電波新聞社
X68000用　5″2HD版　価格未定
★プリンス・オブ・ペルシャ
ブロダーバンドジャパン
X68000用　5″2HD版　価格未定
★パロディウスだ！　コナミ
X68000用　5″2HD版　価格未定
★生中継68　コナミ
X68000用　5″2HD版　価格未定
★遥かなるオーガスタ　ティーアンドイーソフト
X68000用　5″2HD版2枚組　12,800円(税別)
★ノスタルジア　タケル
X68000用　5″2HD版　11,800円(税別)
★ブラックレインボウ　ホビージャパン
X68000用　5″2HD版　8,800円(税別)
★ボンバーマン　システムソフト
X68000用　5″2HD版　7,800円(税別)

FIXER ver.4.0

今月の「大人のためのX68000」は新しいFEP「FIXER ver.4.0」の話。ASKが気に入っている人もいれば，不満点がありながらもしかたなくASKを使っている人もいるでしょう。やはりどんなものにも選択の余地というのは存在すべきですよね。

Ogikubo Kei 荻窪 圭

先月はちょいとばかり時事ネタを扱ったので，今月は馬場ネタを扱う。大腿部骨折だそうである。アナウンサーが選手生命に影響云々をいっていたが，選手として戦っていたこと自体，選手生命を縮めていたようなものだったから，いまさらなにをかいわんや，である。

なんて話はどうでもいい。

そういえば，かのNTTが12月1日からANGEL LINEを始めた。エンジェルである。恥ずかしい名前だ。遊人のANGELにでもあやかろうとしたのだろうか（なににあやかるんだか)。

先立ってANGEL NOTEなる恥ずかしい名前で紫という恥ずかしい色の機械を無料貸し出ししたそうだ。ANGEL NOTEってのはANGEL LINEへアクセスするための単なる端末である。それ以上でも以下でもない。

いい忘れたが，ANGEL LINEというのはNTTが始めたパソコン通信を利用した電話番号検索システムである。ただ，アクセスにはANGEL LINE専用の通信ソフトを使うことになっている。そのソフトだが，PC-9801，J-3100，AXなどMS-DOSマシン用はあるのだが，X68000用はない。いつものこととはいえ，X68000用はない。X1もMZもない。

じゃあダメかというと，なんのことはなくて，パソコンとモデムとそんじょそこいらに転がっている通信ソフトがあれば誰にでも利用できるのだ。

ちなみに，電話番号は0190-104104という変な番号。プロトコルに関しては，1200/2400 bpsというなんの変哲もないものだ。が。ひとつ問題があるとすれば，全角文字のコードが新JISなのだ。シフト新JISではなくて，ただの新JIS。ソフトが対応していればまったくもって，気にすることはない。どうしてシフトJISじゃないのかなあ。ホスト側の機械のせいだろうか。

まあ，いいや。それにしても，104が機械の声になって，有料化したと思ったら，今度はパソコン通信で番号案内ときたもんだ。私はあの，機械の声の104は苦手で苦手で。知らない人のためにいっておくと，いまの104はかけるとお姉さんが出て，こっちが言ったところの番号を探してくれるのだが，見つけたら，「はい，どうぞ」とかいって，機械の声が延々と電話番号を繰り返し言い続けるのだ。おお，不気味。さっさと電話を切ればいいのだが，最初からテープが回っている天気予報や時報ならともかく，始めは人の声なのに唐突にテープかICか知らないが機械の声になって，相手がしゃべりつづけているなか，受話器を置くというのは非常にいやな気分がする。

というわけで，ANGEL LINEは便利だ。人間相手より融通はきかないが，こちらも機械的に応対できるから気が楽。企業か，官公庁か，個人か，フリーダイヤルかで選べる。ただ，プライバシーの問題か，「個人は5件，それ以外は50件に絞られたときに」表示してくれるそうで，ありふれた名前だったりすると，かなり細かいところまで入れてやらないとだめ。また，電話帳に載っている人しかだめだから（当たり前）私の電話番号は検索できるが（ついでに住所まで！），街でナンパした女の子なんかは電話帳に載せてないケースが多いので，たぶんだめだろう。あ，いっておくけど，私は「荻窪圭」なんて名で載っているわけではないので，探さないように。

所在地がある程度絞れないとだめな点を除いて，もうあの電話帳はいらないというメリットは大きい。料金は3分10円で，深夜23時以降は4分10円と，すばやく探せば1件30円の有料化104より安いのも魅力だ。モデムを持っている人はお試しあれ（詳しくはNTTに聞いてください)。

ちなみに，私はNTTの回し者ではないぞ。ただ，面白いと思ったから紹介しただけなのだ。誤解しないように。私の基本姿勢は「NTTは諸悪の根源で，人のコミュニケーション願望につけ込んだ商売をしていて，あの前時代的な電話料金体系は人々を馬鹿にしている」というものだ。

さて，この調子でJRが時刻表をパソコン通信で検索できるようにしてくれると，とてもありがたいな。それいけ，JR。

ああ，関係ない話でこんなに行を稼いでしまった。

FIXER ver.4.0が来た

と，先々月の予告どおりにKamikazeの第3弾をお送りする予定だったのだが，突如届いた1本のソフトのおかげで，またもや予定が狂ってしまった。

その届いたソフトがこのFIXER。今回の「大人のためのX68000」はFIXER ver.4.0のレビューだ。

待っていた人は待っていた。だってASKって馬鹿なんだもん。しょっちゅう同じ間違いを犯すし，何度教え込んでも文法解析が変なのかなんなのか，“浅い”って打ちたいのに，“浅井”が最初の候補になるくらいだから。FIXERも最初は“浅井”だったが，2度目からはちゃんと“浅い”になった。当たり前だけど，偉い。

編集部にこのFIXERが到着して最初に驚いたのが，「ちゃんと学習する」ことだった。というより，学習するとはこういうことをいうのか，っていう感心だ。

たとえば，「庭に埴輪鶏がいる」ってなっても，「庭には二羽鶏がいる」って一度学習させてやると，「庭には二羽鶏がいる」って

ならなかったけれど，２度学習させたら，なった。すごいすごい。

えっと，これは単語の学習と文節内の学習と，文節区切り学習など多くの学習をする関係だと思う。庭には二羽鶏がいる。うーん，完璧。

この賢さは触ってみないとわからないのだが，ASKはもちろんのこと，ATOK7よりもVJE-βよりも賢いのは確かだ，といっていいのではないだろうか。もっとも，ある程度学習させてからの話だ。

FIXERのキー配置

そんなFIXERだが，日本語FEPの善し悪しは変換効率だけで決まるのではない。たとえば，操作性であり，ファイルの大きさであり，アプリケーションとの相性だ。まず，操作性から見ていこう。

一応ASKに準じたキー配置をうたっているが，本当にそうだろうか。

結論からいおう。そうではない。

1)　XF1,XF2,XF3キーについてはシフトともども似ているが，候補がなくなると自動的に文節を伸長してくれるなどめんくらうこともある。スペースが次候補してくれないのも異なる点だ

2)　XF4に関してはさらに異なる。一度押すごとに，かな→カナ→カナ（半角）を繰り返すのだ

3)　ASKでは（たいていのFEPはそうだが）変換中に次の文字を入力すると，未確定文字列は確定され，新しく入力した文字が変換対象になる。FIXERでは確定するキーを押さないかぎりどんどん未確定文字列が増えていく（MAX80文字)

4)　ASK ver.2.0はある程度キーコンフィギュレーション可能だった。FIXERではキーコンフィギュレーションを変えることはできない。ファンクションキーの役割も固定だ（CTRL＋Fキーに収まっている）

5)　CTRLファンクションがまったく効かない。困った。CTRLファンクションを使いまくるどころか，ホームポジションから指を離さずに変換できるよう設定している私は，非常に困る。CTRL＋うんちゃらは，みな全文節確定とされてしまう。思わずCTRL＋Hを押して，泣いたことがある。改良していただけるとうれしい

まあ，慣れの問題かもしれない。

ASKにはなかった機能もいくつかある。

6)　CTRL＋XF2,XF3,XF4,XF5の４つ。CTRL＋XF2が記号入力。これがまたすごい。JISコード.000から部首が登録してあり，部首変換ができるのだ。CTRL＋XF3,XF4,XF5はそれぞれ，かな，ローマ字，CAPS LOCKのON/OFFになっている

7)　SHIFT＋ESC。これは直前に変換した文字列を再変換するというキーである。確定直後のみ効くUNDO技だ

以上が，ASKとは異なる操作性だ。が，まだまだ言葉ではいいづらい違いがいろいろある。

8)　ローマ字に変換できない読みのとき，ASKだとアルファベットのまま残してくれたが（WPなどを除く），FIXERはそういうことをしてくれない

9)　カーソル位置変換にしたとき，変換作業の結果（文節伸長）がカーソル位置に反映されないのは変だ

こうやって比べてみると，ASKより使いづらそうだが，それはASKに慣れているからで，FIXERにしてみれば，ただXFキーに対応しただけにすぎないのだろう。しかも，そのASKへの慣れというのも，けっこうASKの使いづらさをごまかすための対症療法だったりするので，FIXERに文句をいうのは筋違いなものもありそうだ。

ただし，客観的に見て，次の点は許しがたいものがある。いくら賢くてもこれでは使う気にならない，というユーザーもいるだろう。

a.　CTRLファンクションが確定キーになること。コンフィグできるほうがいいのだが，だめだとしても，

・あらかじめCTRL＋H，CTRL＋Mとカーソル移動くらいはCTRLファンクションに割り当て，あとのキーは押してもなにも起こらないようにする

・もしだめなら，CTRLファンクションを一切受けつけないようにする

b.　カーソル位置変換にしたとき，変換作業は反映されないケースがある。これは困る。カーソル位置エコーにした意味がない

このあたりに対するバグフィクス，じゃなくてバージョンアップを早急にしてほしいと思う。

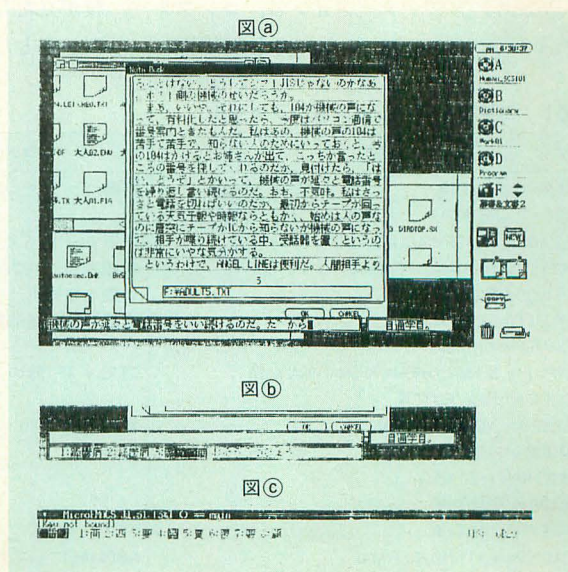
ファイル，辞書の比較

賢いと聞くと，さぞやメモリを食うだろう，と思うのが人情。そこで，以下の表をご覧ください。

ファイルサイズ表

FEP本体	標準システム	65Kバイト
	スモールシステム	63Kバイト
辞書	標準	816Kバイト
	ミドル	636Kバイト
	スモール	454Kバイト

FEP本体である。FIXER.SYSとFIXERS.SYSがあるが，これは後者が機能縮小版である。といっても，カーソル位置での変換ができないだけのようだ。どちらにしろ，ASKよりずっと小さく感じるが，実は気のせいだったりして，FIXERのほうが，若干メモリを多く占有するようだ。ただし，FIXERは辞書バッファの大きさを指定することができるため，その値によって大きく占有メモリは変わる。

辞書はもうとんでもなく大きい。が，小さい辞書もあるので，辞書ドライブの具合によっては，そちらを使う手もあるだろう。辞書メンテナンスはASKについてくるDICMより使いやすいため，自分の辞書を作るのは楽だ。このレビューは大きな辞書で書いている。

なお，この辞書はASK同様，読みに記号も使えるし，単語に半角文字も登録できるので，なんでも登録できて便利だ。

FIXERの辞書は郵便番号も持っているが，これはすべて“/＋番号”という読みで登録されている。

コンフィギュレーション

さて，図1がFIXERのコンフィグファイルだ。ASK2.0より項目が少ないが，これはキーコンフィギュレーションができないからにすぎない。

重要なのは，EXDISPだ。EXTENDかNORMALかで大きく変わる。それは次の項を読めばわかる。

ちなみに，SPEEDは高速にすると速くなるが，変換効率は落ちる。確かに，通常速度では遅いと感じることもあるが，ASKのように辞書を延々と読みにいくことはないので許せる。

PERIODは句読点変換。

学習機能も3種類選べるが，自動的に学習するモードだと変換のたびに学習してくれる。ASKのようにFEPをON/OFFするたびに学習結果を書きにいって，待たされることもない。

動かないアプリ

さて，いろいろと賢くて気に入っているFIXERだが，問題点があったりする。それは図2の表を見れば一目瞭然だ。

左から，アプリケーション名，コンフィギュレーションファイルのEXDISP＝EXTENDのときの動作，NORMALのときの動作だ。とりあえず調べたのはこれだけだ。

実に動かないアプリケーションが多い。EXDISPはNORMALにしよう。EXTENDだと「変換ウィンドウを下に開くアプリケーションのときに表示がおかしくなる」のだ。

それでもワープロ系ソフトとKamikazeはだめだった。SX-WINDOWはまったくだめというわけではなく，変換ウィンドウは開くのだが，動作がおかしくなる。どちらがおかしいのかはわからないが，どちらかのバグかもしれない。VS.Xでは問題なく動くことだし。

WP.Xはまったくだめ。Kamikazeも変換モードにいってもくれない。Hyperwordにいたっては，「ASKバージョン2が組み込まれていません」といって，立ち上がってもくれない。

DOSの世界では，日本語FEPに対応するのはアプリケーションの仕事なので，FIXERが悪いというのも酷だが，ワープロ系ソフトが全滅というのは痛い。早くFIXER対応になってもらいたいものである。

FIXERを使って

いまFIXERを使っているのだが，変換効率だけはやたらといいので，学習さえさせてしまえば，非常に気が楽だ。なんといっても，句読点変換をしていて腹が立たないのだ。逐次自動変換でも腹が立たない。これはすごいことだと思う。使うときは，辞書先読み連文節変換＆句読点変換で，学習優先モードがいい。さすがに逐次自動変換はちょっと怖いからだ。

ただ，X68000の設計の問題で無理なのかもしれないが，ローマ字入力の際，かな→アルファベット変換がほしかったと思う。私などが書くこういった文章には英単語が頻出するので，いちいちローマ字のON/OFFはしたくないのだ。

そういうわけで，Hyperwordが好きな私も，今回だけはあきらめてmicroEMACSを使っていたりする。

気持ちは「頑張れFIXER！」だ。PC-9801ユーザーはNECのMS-DOSについてくるNEC AI変換を使ったりはしない。使いにくいからだ。Macユーザーは，漢字TALKについてくる2.0変換を使ったりはしない。使いものにならないからだ。DynaBookユーザーはATOK7を使う。辞書をROMに持っているからだ。X68000ユーザーはASKを使う。ほかにないからだ。

X68000ユーザーだって，賢いFEPを使いたい。とりあえずは，表にあるFIXERが動作しないアプリケーションをFIXER対応にしてもらって，それからだ。

少なくとも私は，ASKからFIXERに乗り換える心の準備とハードディスクの準備はしているぞ。早いうちにここであげつらった欠点を直してもらいたい。

いつもやっているので，またあてにならない予告をしておくと，来月は確定申告にX68000を使う話でもするとしよう。

図1　コンフィギュレーションファイル

項目	コメント
BUFFRS＝1〜249	辞書バッファ。ひとつ2Kバイト
DIC＝　辞書のファイル名	
LEARN＝AUTO/MANUAL/OFF	学習モード
STYLE＝NORMAL/READ/AUTO	変換方式。AUTOが逐次自動変換
PERIOD＝ON/OFF	句読点変換
PRIORITY=LEARN/MEAN	学習結果と意味解析の優先順位
CHARA＝ZEN/HAN	
INPUT＝ NORMAL/HROMA/KROMA/HIRA/KATA	CTRL＋XF1時の入力モード
CODE＝ HEX/KUTEN	
SPEED＝NORMAL/HIGH	変換速度
MODE＝ SCREEN/LINE	変換位置
BSCHR＝1/2	
CAPS＝ ON/OFF	
FKEY＝ ON/OFF	
EXDISP＝EXTEND/NORMAL	NORMALにしておくべき

図2　対応アプリケーション

	EXDISP＝EXTEND	EXDISP＝NORMAL
ビジュアルシェル	○＊1	○
ED	○	○
WP	○	○
Communication PRO	×＊2	○
CARD PRO	○	○
Kamikaze	×	×
Hyperword	×	×
CyberNote PRO	○	○
Z'sSTAFF PRO	○	○
Music PRO	○	○
CANVAS PRO	○	○
SX-WINDOW	×	×＊3
Magic Palette	○	○
File Processer	○＊1	○
microEMACS	○	○
Muterm	×＊2	○

＊1：変換位置がずれる。
＊2：変換はするが表示されない。
＊3：表示はいいが、変換されない。
なお，microEMACS，Mutermは市販品ではない。

SX-WINDOW

SX-WINDOW上のアプリケーションを作成するための第一歩はSXシステムの仕組みとさまざまな作法を知ることでしょう。また実際のプログラムは，SXシステムがサポートするサブルーチンSXコールを利用することで実現されます。前回の特集ではこのあたりの概念とアセンブラによるプログラミングへの導入を行いました。解説のもとになる資料は先月の付録ディスク「謹賀新年 PRO-68K」に収録されています（169ページを参照）。

さて，SX-WINDOW用プログラムの開発環境としてはアセンブラまたはC言語が中心になると考えられます。そこで今回は，前回の解説をもとにXC ver.2.0でプログラミングを行う場合について見ていくことにしましょう。また，あわせてウィンドウ上のプログラムで皆さんの関心が高いと思われるグラフマネージャの使い方を解説します。基本的な図形の表示をもとに理解を深めてください。

¥WORKS¥CONTENTS

CONTENTS

ウィンドウプログラミングへの道（2）

C言語によるプログラミング

Murata Toshiyuki　村田 敏幸

SX-WINDOW上のプログラムを作成するための基礎知識として，SXシステムのしくみを解説し，同時にアセンブラによる基本的なプログラミングの作法を示しました。今回は前回の話をベースにC言語で開発を行う場合について解説しましょう。

先月に引き続き，SX-WINDOWプログラミングの基礎編です。今回はC言語によるプログラムの作り方を解説し，さっと切り上げます。用語などにつきましては，適宜，前回の記事を参照してください。また，Cでプログラム開発を行うための道具として，XC ver.2.0と先月号の付録ディスクに収録されていたヘッダファイルとC用のライブラリを用意してください。

なお，不本意ながら，この記事はXC Ver.2.0のみの使用を前提に進めます。先月号の付録ディスクで配布されたヘッダファイル，ライブラリはXC Ver.2.0用で，少なくとも，XC Ver.1.0，ないしは，gcc＋XC Ver.1.0のライブラリでは使えません。また，gcc＋XC Ver.2.0のライブラリでの開発はメモリ不足で，ほとんど実用にならないでしょう。2Mバイトのメインメモリでは，あとで示す小さなサンプルプログラムをコンパイルするのが精一杯です（当然，ASK68Kなど不要なデバイスドライバはすべて外していますし，MAKEも使っていませんが）。

原因は，SX-WINDOW用のヘッダファイルの中で，多くのデータ型・関数が一括して定義・宣言されているためです。ヘッダがマネージャ別に分割してあり，必要なものだけをインクルードできるようになっていれば，おそらくgccも利用でき，XCを使うときにもコンパイル時間が短縮されるのですが。

さっそく環境作りを

では，SX-WINDOWの開発キットをCプログラミング環境に取り込んでおきましょう。先月号の付録ディスクを展開して得られた4枚目のディスク中，¥SX¥INCLUDEにあるヘッダファイルSXLIB.H，SXDEF.HをXC標準のヘッダファイルを収めたディレクトリに，¥SX¥LIBにある__ライブラリファイルSXLIB.AとMAINR.OをXCのライブラリと同じディレクトリに，それぞれ，コピーします。また，¥SX¥TOOLにある4本のファイル中，少なくともWDB.Xは，パスの通った適当なディレクトリに置いておきます（ディスクにゆとりがあるようでしたら残りの3本も一緒に転送しておきましょう）。

さらに，CONFIG.SYSにも注意を向けます。メモリの増設具合にもよりますが，FSX.XをCONFIG.SYSで組み込んでしまうと，コンパイル時のフリーエリアが十分確保できない可能性があります。FSX.Xは動作試験時など必要なときにのみ組み込み，用がすんだらすかさず外すようにします。

動作試験といえば，SXシェルが起動時に無条件にG-RAMを初期化するのを忘れてはいけません。G-RAMをRAMディスクにして，その上で開発を進めていたりすると，動作試験を行った瞬間にRAMディスク中のファイルがみんな消えてしまいます。G-RAMはRAMディスクとしては使わないようにするか，せいぜいテンポラリディスクとして使うにとどめましょう。もっとも，SXシェルがG-RAMを初期化するのに利用しているIOCSコール90_HのG_CLR_ONを殺してしまう技もないではありませんが（どこかで，そんな公開ソフトを見掛けたことがあるような気もします）。

先月のサンプルをCで書き直す

作業環境さえ整えれば，もうすぐにでもSX-WINDOWプログラミングにとりかかることができます。プログラムの構造そのものは，先月お話ししたアセンブリ言語の場合となんら変わりありません。というわけで，さっそくですが，サンプルプログラムをリスト1，そのコンパイル用のバッチファイルをリスト2に示します。リスト1は，先月，アセンブリ言語で書いたWINTEST.SをCで書き直したもので，両者の動作はまったく同じです（なんてったって，ハンドディスコンパイルしたんですから）。ウィンドウを開き，適当に文字列を表示します。

以下，プログラムの各部を順に見ていきますが，スペースの都合もあり，余り詳しくは説明できませんので，足りない部分はリスト中のコメントと，先月のアセンブリ言語版の解説で補ってください。

●大域変数の定義（46〜51行）

このプログラムはリエントラントに書いてあります。先月も触れたように，プログラムをリエントラントに書いておくと，同時に複数起動するときでも，プログラムのコード自体はメモリ上にただひとつあればすみます。同じプログラムをいくつもメモリに置く必要がない分，メモリの使用量を抑えることができるというわけです。メモリは貴重な資源ですから，SX-WINDOW上では“プログラムはリエントラントに作る”のが基本です。

リエントラントに書かれたプログラムでは，プログラムの実行コードだけではなく，静的に宣言された変数も複数のタスクから共用されることになります。各タスクに固有の変数は，必ずauto変数として，スタック上に用意しなければなりません。

ここで，Cの言語仕様上の問題が顔を出します。どうやって大域変数を実現するかという問題です。リエントラントに書くためには，関数の外部で変数を宣言するわけにはいきませんし，といって，auto変数はひとつの関数内でしか参照できない局所的な変数です。Cではスタック上に用意した変数を複数の関数から直接参照する手段が用意されていません。

ひとつの解決策がリストの46〜51行です。“本当は関数の外で定義したかった変数”をみんなひっくるめて構造体でくくり，GVALという名前のデータ型としてtypedefしてあります。そして，ずーっと下，メインルーチン内の291行で，この型をもったデータオブジェクトを実際にスタック上に確保し，以下，関数呼び出しのときには，この構造体へのポインタを関数へ渡してやるようにします（たとえば，300行の

ように)。呼び出された関数側では，この構造体へのポインタに“－＞”演算子を適用することで，個々の“疑似大域変数”にアクセスできるというわけです。

なお，こうして実現した“疑似大域変数”は，あくまで，auto構造体ですから，明確な初期化（291～294行）を行わない限り，初期値は不定です。ふつうの大域変数感覚で初期化をサボると，ときに，プログラムが異常動作することになります。

●メインルーチン（289～325行）

アセンブリ言語でSX-WINDOW用プログラムを書いたときには，COMMAND.Xから直接起動する場合と，SXシェル上で起動する場合との2つのエントリを用意し，それぞれ，ある程度の初期化動作を行う必要がありました。しかし，Cの場合は，このあたりの決まりきった処理はスタートアップルーチン（Cプログラムに必ずリンクされ，諸々の初期化を行うモジュール）が肩代りしてくれます。関数mainでは純粋にプログラム自身の初期化（ウィンドウを開くとか，変数を初期化するとか）から始めればよいのです。リスト1では，300行でウィンドウを開く関数initを呼び出し，エラーなくウィンドウが開けたら，イベントドリブン型プログラムの特徴的なメインループ（304～324行）になだれ込んでいます。

●初期化部（80～108行）

初期化部分には，わずかにアセンブリ言語版と異なる点があります。84行と86行がC使用時特有の処理です。

84行では，TSSetAbortにより，ハードウェアエラーにより中断された場合の処理関数を登録しています。このTSSetAbortはSXコールではなく，C独自の関数です。

先月のアセンブリ言語版ではSXコールTSEventAvailを呼び出したときに，その戻り値を調べ，エラーの有無を調べたわけですが，ライブラリ関数TSEventAvailは関数内部でこのハードウェアエラーを調べ，エラー時にはTSSetAbortで指定された関数に自動的に制御を移します。TSSetAbortは2つの引数をとります。第1引数は中断時に実行される関数へのポインタです。また，第2引数はその中断時処理関数に渡される引数です。通常，例の“疑似大域変数”へのポインタを指定することになるでしょう。

86行ではTSGetTdbにより，82行で用意したtask構造体に自分自身のタスク管理ブロックを読み込んでいます。これは，続く89行のTSTakeParamに渡す“argvに展開される前のコマンドライン文字列”を得るためです（コマンドライン文字列はタスク管理ブロック中に格納されています)。アセンブリ言語レベルでなら，タスク起動時にa2レジスタで渡されたコマンドライン文字列がそのまま利用できるのですが，Cではここでやったような回りくどい手段をとる必要があるのでした（実はもう少し深い意味があるのですが，端折ります)。

89行以降は，先月のアセンブリ言語版と変わりありません。ウィンドウを開くときの常套手段です。89～95行でウィンドウの表示位置を決め，98～99行のWMOpenによってウィンドウを作成します。WMOpenは，ウィンドウが作成できた場合には作成したウィンドウのウィンドウ構造体へのポインタを，また，メモリ不足などの理由で作成に失敗したときにはNULLを返します。NULLが返った場合には，もうどうしようもありませんから，メインルーチンに戻って，そのまま終了です。うまくオープンできた場合は，必要に応じてウィンドウの細かな初期化を行います。

既存のデスクアクセサリの多くでは，ここで描画色や背景色などの設定を行っています。もっとも，ウィンドウを開いたときに，描画色は黒（正確には少し違うけどパス)，背景色は明るいグレーというように，かなり都合よく初期化されますから，リスト1ではばっさり省略してあります。

●マウス左ボタンダウンイベント発生時の処理（113～141行）

マウスの左ボタンはアクティブウィンドウの切り替えやウィンドウの移動・拡大に使うよう決められているボタンですから，ここで，ウィンドウに対する操作の大部分を処理します。ほとんど決まりきった手順ですから，多くの場面でリスト1がそのまま利用できるでしょう。基本的なウィンドウ操作以外のプログラム独自の処理が必要な場合には，133行以下のswitchの中に，

```
    case W_ININSIDE:
        処理
        break;
```

のような形でウィンドウパートコードに応じた処理を追加していくことにになります。

●アップデートイベント発生時の処理（230～251行）

アップデートイベントについては，先月も触れましたが，若干言葉が足りなかったようですので，この場で補足します。

下敷きになっていたウィンドウが上になり，いままでほかのウィンドウで隠されていた部分が見えるようになった場合，ウィンドウマネージャは新たに見えるようになった領域を，対応するウィンドウのアップデートリージョンにつけ加えます。イベントマネージャはほかに重要なイベントがない場合に，各ウィンドウのアップデートリージョンを調べ，もし空でなければ，画面再描画の必要ありと判断して，該当ウィンドウにアップデートイベントを送ります。

アップデートイベント発生時の処理手順は次のようになります。

1)　そもそもイベントが自分に向けられたものかどうか調べる（233行)。そうでなければなにもしない。

2)　WMUpdateを呼び出し，以後の描画がアップデートリージョンでクリッピングされるようにする（235行)。

3)　画面を描き直す前に，必ず，GMSetGraphを使って，自分をカレントグラフにする（238行)。

4)　実際に画面を描き直す（リスト1では，146～228行の下位関数で実際の描画を行っている)。すでにアップデートすべき範囲でクリッピングしてあるから，何も考えずに全画面を描き直してよい。

5)　再描画がすんだら，WMUpdtOverを呼び出し，WMUpdate呼び出し前のクリッピング範囲に戻す（248行)。

ここで，もし，アップデートリージョン以外の領域も同時に描き直したい場合には，次のいずれかの方法をとります。

1)　WMUpdate呼び出しに先立って，WMAddRectやWMAddRgnを使って，再描画したい領域をアップデートリージョンに加えておく。

2)　WMUpdate呼び出し後に，WMClipRectやWMSetClipによりクリッピング範囲を再設定する。

3)　WMUpdateを呼び出す前，あるいは，WMUpdtOverを呼び出したあとに不足部分を描画する（リスト1ではこの方法でグローボックスを描いています)。

どの場合にも，WMUpdateとWMUpdtOverは必ず呼び出さなければなりません。これらのコールにより，アップデートリージョンが空になり，“イベントマネージャがアップデートずみだと判断できるようになる”からです。これらコールを呼び出さない限り，イベントマネージャは永久にアップデートイベントを発行し続けることになり，プログラムが止まってしまいます。

●アクティベートイベント発生時の処理（256～265行）

アクティベートイベントはウィンドウがアクティブになったときに，最優先で発行されます。たとえば，ウィンドウを新たに

開いたときや，WMSelectで自分のウィンドウをアクティブにしたときには，その直後にアクティベートイベントが発行されることになります。

ここでやるべきことは，“自分がいまアクティブかどうかを内部のフラグに覚えておく”ことだけです。既存のデスクアクセサリの中には，“自分がアクティブかどうかのフラグ”をWMSelectを実行したときなどにも更新している場合があるようですが，フラグの更新は，アクティベートイベント発生時に一括して行えば十分です。

●タスクマネージャからのイベント発生時の処理（270〜284行）

タスクマネージャからのイベントの具体的な指示内容は，イベントを取得したときにイベントレコードのwhat2フィールドに格納されていますから，その内容に応じ，switchで処理を振り分けます。リスト1では，どんなプログラムでもサポートしなければならない最低限のイベントのみを処理対象にしています。

●終了時の処理（61〜67行）

ウィンドウを閉じ，終了しているだけです。本来，SX-WINDOW上のタスクはTSExitで終了するのですが，SX-WINDOW用CライブラリにはTSExitはなく，ふつうどおりにexitを使う約束になっています。

注意すべきポイント

最後に，CでSX-WINDOW上のプログラムを作る際の注意点を挙げておきます。

・メモリ消費を抑えるために，スタックサイズとヒープ（SX-WINDOWのヒープではなく，mallocで確保して使うCのヒープ）サイズはなるべく小さくとる。

XCでは，特に指定しないとスタックとヒープにそれぞれ64Kバイトのメモリを確保します。2Mバイト程度のメインメモリでSXシェルを使っているときに計128Kバイトものメモリを占有するのはほとんど犯罪行為ですから，/Gs，/Ghの各スイッチで，必要な分だけ確保するようにしましょう。スタックを小さくすることに不安があるのであれば，スタックチェックのコードの生成を指定する/Gcスイッチをつけてコンパイルし，スタックがあふれていないかどうか確認するようにします。なお，スタックサイズ，ヒープサイズの最小値は，それぞれ，4Kバイト，8Kバイトです。

・画面への出力にはC標準関数ではなく，SX-WINDOWのライブラリを使う。

いまのところ，printfなどのC関数はSX-WINDOWに対応していません。もし，printfに相当する処理を行いたい場合には，sprintfで一度char型の配列に出力文字列を作成し，その後，GMDrawStrZで表示する，といった細工が必要です。

・Cの低レベルI/O関数はなるべく使わない。

クリーナーの関係で，SX-WINDOW上でのファイルI/Oには，DOSコールではなく，SXコールを使う約束になっています。一応，基本的な入出力は一般的なCプログラムと同じ手順で行えるよう，SXLIB.A中にはSXコールに対応したopenとcloseが用意されてはいますが（これらは間接的にfopen，fcloseからも呼び出されます），ファイルの属性を取得・設定するとか，ファイルを消去するとか，ディレクトリ中から特定ファイルを検索するといった関数に対してはなんの配慮もありません。これらの処理が必要なときには，タスクマネージャのSXコールから同等のものを探してきて置き換える必要があります（たとえば，remove→TSDeletePというように）。

・（Cの）ヒープのサイズを変更する関数は使用できない。

SX-WINDOW上のCプログラムでは，（Cの）ヒープは（SX-WINDOWの）ヒープ上に確保されます。そのため，sbrkやbldmemといったヒープを拡張する関数は使用できません。ヒープが足りなくなったら勝手にsbrkするようにmallocを改造している人も多いと思いますが，SX-WINDOW上では通用しませんから注意してください。

・リエントラントなプログラムでは，Cライブラリ関数内の静的ワークにも配慮する。

Cの関数の中には，内部に静的なワークを持っているものがあります。たとえば，fopenなどで使われるファイル構造体の配列とか，mallocなどで使われる（Cの）ヒープ管理用ポインタ，strtok内の小規模なワークなどです。リエントラントなプログラムでは，これらの目に見えないワークも複数のタスクで共有されます。

SX-WINDOWでは，さほど厳密な意味でのリエントラントさは要求されませんから，内部にワークをもったライブラリ関数が即使用不可というわけではありませんが（TSEventAvailを呼び出すまでは，ほかのタスクが同じライブラリ関数を使用することはありえないから安全），ワークの競合には一応気をつけなければなりません。

致命的な例としてはfclosealllがあります。fcloseallによりファイルを全部まとめてクローズしようなんてことをすると，コードを共有する全タスクのファイルがまとめて閉じられてしまうのです。

・リエントラントなプログラムでは，オープンできるファイルの数や，使用できる（Cの）ヒープ領域がいつもより少ないことを計算にいれる。

上で述べたこととダブりますが，リエントラントなプログラムではファイル構造体を格納しておくライブラリの内部ワークや，（Cの）ヒープも，複数のタスクで共用されます。自分はまだ，mallocを1度も呼び出していないのに，すでにヒープに空きがないという事態も起こりうるわけです。再三，SX-WINDOW上ではプログラムは可能なかぎりリエントラントに作るよういってきましたが，この制限を回避するためには，リエントラントに作ることを諦めるしかありません。

＊

以上，CによるSX-WINDOWプログラミングの基本部分について，簡単にまとめてみました。

リスト1　WINTEST.C

```
 1: /*
 2:           ウィンドウを開いて文字を表示する
 3: */
 4: #define NULL          ( ( void * ) 0 )
 5: #include    <stdlib.h>
 6: #undef   __POINT_T   /*point_t型を間違いなくlong型と等価にする呪文*/
 7: #include    <sxlib.h>
 8:
 9: typedef enum {  FALSE, TRUE, } BOOLEAN;
10: #define G_PLAIN       0              /*  プレーンな書体  */
11:
12: /*  x,y座標からpoint_t型を作るマクロ     */
13: #define Pt(x,y)       ( ( point_t )( ( point_t )(x) << 16 | ( short )(y) ) )
14:
```

```
 15: /*  point_t型をx,yに分けるマクロ     */
 16: #define PtX(p)        ( ( short )( (p) >> 16 ) )
 17: #define PtY(p)        ( ( short )(p) )
 18: #if 0
 19:     /*  別案（こっちは変数専用）*/
 20:     #define _PtX(p)       ( *( short * )&(p) )
 21:     #define _PtY(p)       ( *( ( short * )&(p) + 1 ) )
 22: #endif
 23: /*
 24:     WDEFID        ウィンドウ定義関数ID（標準ウィンドウ）
 25:     WINOPT        ウィンドウオプション（グローボックスあり＆ON）
 26:     WINOPTLOW     ウィンドウオプション下位4ビット
 27:     WINDEFID      ウィンドウ定義ID
 28:     WINH          ウィンドウ横ドット数
 29:     WINV          ウィンドウ縦ドット数
 30:     WINSIZE       ウィンドウの大きさ  point_t
 31:     WINTITLE      ウィンドウタイトル  LASCII *
 32: */
 33: #define WDEFID        WI_STD
 34: #define WINOPT        ( WC_GBOX | WC_GBOXON )
 35: #define WINOPTLOW     ( WINOPT & 0xf )
 36: #define WINDEFID      ( WDEFID << 4 | WINOPTLOW )
 37: #define WINH          208
 38: #define WINV          160
 39: #define WINSIZE       Pt( WINH, WINV )
 40: #define WINTITLE      ( ( LASCII * )"¥x08untitled" )
 41:
 42: /*  イベントマスク  */
 43: #define EVENTMASK     ( EM_MSLDOWN|EM_UPDATE|EM_ACTIVATE|EM_SYSTEM1|EM_SYSTEM2 )
 44:
 45: /*  データエリア（大域変数）    */
 46: typedef struct {
 47:     window  *winptr;                    /*ウィンドウ構造体へのポインタ*/
 48:     BOOLEAN activeflag;                 /*自分がアクティブかどうかのフラグ*/
 49:     rect    winsize;                    /*ウィンドウの位置*/
 50:     tsevent eventrec;                   /*タスクマネージャ用のイベントレコード*/
 51: } GVAL;
 52:
 53: /*  デバッグ用シェルを起動するためのコマンドライン  */
 54: char _sxkernelcomm[] = "sxwdb.x -D -K";
 55:
 56: /***********************************************************************/
 57:
 58: /*
 59:     終了時処理
 60: */
 61: void term( GVAL *gp, int retval )
 62: {
 63:     if ( gp->winptr != NULL )           /*ウィンドウをオープンしているのなら*/
 64:         WMDispose( gp->winptr );        /*ウィンドウを破棄する*/
 65:
 66:     exit( retval );                     /*終了コードをもって帰る*/
 67: }
 68:
 69: /*
 70:     エラーによる中断時処理（TSSetAbortで登録する）
 71: */
 72: void myabort( int dummy, GVAL *gp )
 73: {
 74:     term( gp, EXIT_FAILURE );
 75: }
 76:
 77: /*
 78:     初期化（ウィンドウオープン）
 79: */
 80: BOOLEAN init( GVAL *gp )
 81: {
 82:     task taskbuf;
 83:
 84:     TSSetAbort( myabort, ( long )gp );  /*中断時処理関数を登録する*/
 85:
 86:     TSGetTdb( &taskbuf, -1 );           /*タスク管理ブロックを得る*/
 87:
 88:     /*ウィンドウの表示位置を決める*/
 89:     if ( ( TSTakeParam( &taskbuf.command,
 90:         &gp->winsize, NULL, 0, NULL, NULL ) & 1 ) == 0 ) {
 91:         *( point_t * )( &gp->winsize.left )
 92:             = ( point_t )TSGetWindowPos();
 93:         *( point_t * )( &gp->winsize.right )
 94:             = *( point_t * )( &gp->winsize.left ) + WINSIZE;
 95:     }
 96:
 97:     /*ウィンドウをオープンする*/
 98:     if ( ( gp->winptr = WMOpen( NULL, &gp->winsize, WINTITLE, TRUE,
 99:         WINDEFID, ( window * )-1, TRUE, TSGetID() ) ) == NULL ) {
100:         return ( FALSE );               /*オープンできなかった*/
101:     } else {
102:         gp->winptr->wOption = WINOPT;   /*ウィンドウオプションを設定する*/
103:
104:         GMSetGraph( gp->winptr );       /*自分をカレントグラフにしてから*/
105:         WMDrawGBox( gp->winptr );       /*グローボックスを描く*/
106:         return ( TRUE );                /*初期化完了*/
107:     }
108: }
109:
110: /*
111:     マウス左ボタンダウンイベントの処理
112: */
113: void mouseLdown_event( GVAL *gp )
114: {
115:     window *wtemp;
116:
117:     /*ボタンが押された位置が自分のウィンドウかどうか調べる*/
118:     /*（自分のウィンドウ上でなければなにもせずに戻る）*/
119:     if ( ( window *) gp->eventrec.whom == gp->winptr ) {
120:         TSGetEvent( EVENTMASK, &gp->eventrec ); /*このイベントを取り除く*/
121:
```

```
122:         if ( !gp->activeflag ) {            /*いまアクティブでなければ*/
123:             WMSelect( gp->winptr );         /*自分のウィンドウをアクティブにする*/
124:
125:             /*タイトルバーがドラッグされた?*/
126:             if ( WMFind( ( point_t )gp->eventrec.whom2, &wtemp ) != W_INDRAG
127:                 || EMLStill() == 0 )
128:                 return;                     /*そうでなければ即戻る*/
129:         }
130:
131:         /*あとの処理をウィンドウマネージャにまかせて*/
132:         /*ウィンドウ内のどこでボタンが押されたのかという情報だけもらう*/
133:         switch ( SXCallWindM( gp->winptr, &gp->eventrec ) ) {
134:             case W_INCLOSE:                     /*クローズボックス上だった場合は*/
135:                 term( gp, EXIT_SUCCESS );       /*タスクを終了する*/
136:                 break;
137:             default:
138:                 break;
139:         }
140:     }
141: }
142:
143: /*
144:     アップデートイベントの処理
145: */
146: void test1( int fontkind, point_t fontsize, point_t pos, point_t feed )
147: {
148:     GMFontKind( fontkind );                 /*フォントの種類をセット*/
149:     GMFontSize( fontsize );                 /*フォントの大きさをセット*/
150:
151:     GMFontFace( G_PLAIN );                  /*ふつうの書体*/
152:     GMMove( pos );                          /*ペン座標=表示開始位置を指定*/
153:     GMDrawStrZ( "標準(plain)" );            /*文字列を表示*/
154:     pos += feed;                            /*つぎの表示に備えて*/
155:                                             /*ペン座標を進める(改行に相当)*/
156:
157:     GMFontFace( G_BOLD );                   /*以下,同様*/
158:     GMMove( pos );
159:     GMDrawStrZ( "強調(bold)" );
160:     pos += feed;
161:
162:     GMFontFace( G_ITALIC );
163:     GMMove( pos );
164:     GMDrawStrZ( "斜体(italic)" );
165:     pos += feed;
166:
167:     GMFontFace( G_ULINE );
168:     GMMove( pos );
169:     GMDrawStrZ( "下線(underline)" );
170:     pos += feed;
171:
172:     GMFontFace( G_OLINE );
173:     GMMove( pos );
174:     GMDrawStrZ( "袋文字(outline)" );
175:     pos += feed;
176:
177:     GMFontFace( G_SHADOW );
178:     GMMove( pos );
179:     GMDrawStrZ( "影つき(shadow)" );
180:     pos += feed;
181:
182:     GMFontFace( G_ITALIC|G_SHADOW );
183:     GMMove( pos );
184:     GMDrawStrZ( "組み合わせたりもして" );
185: }
186:
187: void test2( void )
188: {
189:     point_t size;
190:
191:     GMFontFace( G_PLAIN );                  /*標準の書体*/
192:     GMFontKind( G_ROM24 );                  /*24ドットフォント*/
193:     GMMove( Pt( 4, 380 ) );                 /*ペン位置=(4,380)*/
194:
195:     /*文字の大きさを8x8ドットから32x32ドットまで変化させながら*/
196:     /*1文字表示を繰り返す(文字は横に並べる)*/
197:     for ( size = Pt( 8, 8 ); size <= Pt( 32, 32 ); size += Pt( 1, 1 ) ) {
198:         GMFontSize( size );                 /*フォントサイズをセット*/
199:         GMDrawChar( 'も' );                 /*1文字表示*/
200:                                             /*(ペン位置は1文字分進む)*/
201:     }
202: }
203:
204: void test3( void )
205: {
206:     point_t size, pos;
207:
208:     GMFontFace( G_PLAIN );                  /*標準の書体*/
209:     GMFontKind( G_ROM24 );                  /*24ドットフォント*/
210:
211:     /*文字の大きさを64x8ドットから64x28ドットまで変化させながら*/
212:     /*1文字表示を繰り返す(文字は縦に並べる)*/
213:     for ( size = Pt( 64, 8 ), pos = Pt( 400, 4 ); size <= Pt( 64, 28 );
214:         size += Pt( 0, 1 ), pos += Pt( 0, PtY( size ) ) ) {
215:         GMFontSize( size );                 /*フォントサイズをセット*/
216:         GMMove( pos );                      /*ペン位置を指定*/
217:         GMDrawChar( 'み' );                 /*1文字表示*/
218:     }
219: }
220:
221: void test4( void )
222: {
223:     GMFontFace( G_PLAIN );                  /*標準の書体で*/
224:     GMFontKind( G_ROM24 );                  /*24ドットフォントを*/
225:     GMFontSize( Pt( 128, 128 ) );           /*128x128に拡大して*/
226:     GMMove( Pt( 220, 160 ) );               /*座標(220,160)に*/
227:     GMDrawChar( '誓' );                     /*1文字表示する*/
```

▶1月号のハガキの質問を見て，僕は狭い部屋の畳いっぱいに12冊のOh!Xを並べ，1時間ずっと本の間を行ったり来たりして，やっと順番を決めた。しかし，Oh!Xの1月号を見ていて，ふと170ページで手が止まった。私の苦労っていったい？　横山　誠(18)群馬県

```
228: }
229:
230: void update_event( GVAL *gp )
231: {
232:     /*自分のウィンドウかどうか調べる*/
233:     if ( ( window *) gp->eventrec.whom == gp->winptr ) {
234:     /*アップデート開始*/
235:         WMUpdate( gp->winptr );          /*クリッピングリージョンと  */
236:                                          /*アップデートリージョンを一致させる*/
237:
238:         GMSetGraph( ( graph * )gp->winptr );     /*自分をカレントグラフにする*/
239:
240:     /*文字を拡大・縮小表示したりして遊ぶ*/
241:         test1( G_ROM16, Pt( 16, 16 ), Pt( 4, 4 ), 20 );
242:         test1( G_ROM12, Pt( 12, 12 ), Pt( 220, 4 ), 20 );
243:         test1( G_ROM24, Pt( 24, 24 ), Pt( 4, 160 ), 28 );
244:         test2();
245:         test3();
246:         test4();
247:
248:         WMUpdtOver( gp->winptr );        /*クリッピングリージョンを元に戻す*/
249:         WMDrawGBox( gp->winptr );        /*グローボックスを描く*/
250:     }
251: }
252:
253: /*
254:     アクティベートイベントの処理
255: */
256: void activate_event( GVAL *gp )
257: {
258:     /*アクティブになったのが自分だったら*/
259:     if ( ( window *) gp->eventrec.whom == gp->winptr )
260:         gp->activeflag = TRUE;           /*フラグをONにする*/
261:
262:     /*アクティブになったのがほかのタスクだったら*/
263:     else if ( ( window *) gp->eventrec.whom != NULL )
264:         gp->activeflag = FALSE;          /*フラグをOFFにする*/
265: }
266:
267: /*
268:     タスクマネージャからのイベントの処理
269: */
270: void system_event( GVAL *gp )
271: {
272:     /*イベントレコードのwhat2で指定される指示に従い処理を振り分ける*/
273:     switch ( gp->eventrec.what2 ) {
274:         case ENDTSK:                     /*タスクを終了しろ!*/
275:         case CLOSEALL:                   /*ウィンドウを閉じろ!*/
276:             term( gp, EXIT_SUCCESS );    /*はい,はい*/
277:             break;
278:         case WINDOWSELECT:               /*ウィンドウをアクティブにしろ!*/
279:             WMSelect( gp->winptr );      /*ほい,ほい*/
280:             break;
281:         default:
282:             break;
283:     }
284: }
285:
286: /*
287:     メインルーチン
288: */
289: void main( void )
290: {
291:     GVAL gval = {                               /*大域変数の代わり*/
292:         NULL,            /*winptr = NULL*/
293:         FALSE,           /*activeflag = FALSE*/
294:     };
295: #if 0
296:     gval.winptr = NULL;                  /*ウィンドウ構造体へのポインタをクリア*/
297:                                          /*(ウィンドウ未オープンの印)*/
298:     gval.activeflag = FALSE;             /*アクティブかどうかのフラグをOFF   */
299: #endif
300:     if ( !init( &gval ) )                /*初期化する*/
301:         term( &gval, EXIT_FAILURE );     /*初期化に失敗したら即終了
302:
303:     /*****メインループ*****/
304:     for ( ;; ) {
305:         TSEventAvail( EVENTMASK, &gval.eventrec );  /*イベントを取得する*/
306:
307:         switch ( gval.eventrec.what ) {  /*イベントの種類に応じて処理を振り分ける*/
308:             case E_MSLDOWN:              /*左ボタンダウンイベント*/
309:                 mouseLdown_event( &gval );
310:                 break;
311:             case E_UPDATE:               /*アップデートイベント*/
312:                 update_event( &gval );
313:                 break;
314:             case E_ACTIVATE:             /*アクティベートイベント*/
315:                 activate_event( &gval );
316:                 break;
317:             case E_SYSTEM1:              /*タスクマネージャからのイベント*/
318:             case E_SYSTEM2:
319:                 system_event( &gval );
320:                 break;
321:             default:                     /*そのほかのイベント*/
322:                 break;
323:         }
324:     }
325: }
```

リスト2 MK.BAT(リスト1のコンパイル用バッチファイル)

```
cc /O /Gs4k /Gh8k wintest.c %lib%¥__mainr.o %lib%¥sxlib.a
```

▶「はてしない物語」の英語版はひどいです。2色刷りじゃないし，ハードカバーじゃないし，表紙は写真だし，挙げ句の果てには広告文まで入れているもんなあ（ド版がほしくなった）。
黒田　昌利(19)東京都

目玉を小さくするプログラム!?

Izumi Daisuke 泉　大介

1月号付録ディスクのSX-WINDOW用プログラムSXeyes, 15パズル, パス名.Xは, 実行時に140Kバイトものメモリが必要で, 1Mバイトのシステムでは2つ以上同時に実行するのが困難でした。patch.basはこれらにパッチを当て, 実行時に必要なメモリを30Kバイト程度にまで小さくするプログラムです。

●使用方法

まずX-BASICを起動し，リスト1を入力してください。30行でdir$にセットしているのは，SXeyes.X，15パズル.X，パス名.Xの入っているディレクトリ名です。自分の環境に合わせてディレクトリ名はセットし直してください。以下に典型的な手順を示しますので，X-BASICの起動方法がわからない方は参照してください。

まず,付録ディスクを解凍して得られるDISK3をドライブ0にセットし,リセットします。これでVS2.Xが起動します。続いてドライブ1に本体に付属してきたシステムディスク(のコピー)を挿入してください。BASIC2と書かれたアイコンがありますね。これをダブルクリックすると，X-BASICのディレクトリが開きます。この中にBASIC.Xという名前のファイルアイコンがあります。これをダブルクリックするとX-BASICが起動します。

X-BASICが起動したらリスト1を入力します。「10 /＊………」と1行入力したらリターンキーを押してください。これでプログラムが登録されます。同じ要領でプログラムの続きを1行ずつ順に入力していきます。なお，最初の2行（10……，20……の2行）はプログラムの注釈行ですので，入力しなくてもかまいません。また，3番目の行はこの操作を想定して作成してあるので書き換える必要はありません。

プログラムの入力が終わったら，間違いがないか十分チェックしてください。特にfseekがあるところは入念に。ここを間違えるとディスクの解凍からやり直す羽目に陥ります。

チェックしてOKならプログラムを実行します。「run」と入力してください。すぐに実行は終了します。これで2つのファイルの実行時に必要なメモリは小さくなりました。1MバイトのSX-WINDOWでも，数個の目玉がマウスカーソルを追いかける不気味さを体験することができます。

●パッチ当てプログラムの理論

これらのサンプルプログラムは，スタックとして64Kバイト，ヒープとして64Kバイトのメモリがデフォルトで用意されます。つまり起動するだけで単純に128Kバイトものメモリが持っていかれてしまうのです。目玉が140Kバイトものメモリを喰ってしまう原因はここにあります。これをなんとかすれば目玉を小さくできるはずです。

そう考えてSXeyes.Xをデバッガで覗いてみると,図1のような部分がありました。プログラム内の特定の場所からデータを取り出してチェックし，再び同じ場所に戻しています。参照している$137AF4, $137AF8の2つのアドレスには,$10000つまり64Kバイトがセットされており,いかにも怪しそうです。サンプルプログラムにはシンボルテーブルがないので確信がもてず,SX-WINDOW用の簡単なプログラムを作ってデバッガで覗いてみると,同じ処理を行っている部分がありました。こちらには__STACK，__HEAPというシンボルがふってあり間違いなさそうです。

あとは参照している2つのアドレスの，プログラム先頭からのオフセットを出し，そこに適当な値を書き込むだけでOKです。こうして作ったのがリスト1のプログラムです。リスト1では1024を__STACK，__HEAPに書き込んでいます。実行時には図1の部分でスタックは4Kバイト，ヒープは8Kバイトにセットし直されます。これで128Kバイトを12Kバイトにまで小さくすることができました。

__STACK，__HEAPの値は，図2のように使われています。メモリマンに要求して確保したヒープのサイズを変更するときに使用するようです。3番目に加えている数値は意味不明ですが，おそらく親プロセスの環境変数エリアのサイズではないかと思います。時間がなく，これは詳しく追いかけることができませんでした。

スタック・ヒープサイズを勝手に変更してしまって大丈夫なのだろうかという不安が残りますが，私が動作チェックをした範囲では問題なく動いています。いずれもヒープを使用するプログラムとは思えませんし，4Kバイトのスタックサイズを使い尽くしてしまうような処理も行っているようには見えません。どうぞ，目玉を満喫してください。

リスト1　パッチ当てプログラム

```
10 /* SXeyes、15パズルのある
20 /*  ディレクトリ名を に書き込む
30 str dir$ = "A:¥SX_SAMPLE¥"
40 int fp
50 int code( 0 ) = { 1024 }
60 /*
70 fp = fopen( dir$+"SXeyes.x", "w" )
80 fseek( fp, &H1B44+64, 0 )
90 fwrite( code, 1, fp )
100 fwrite( code, 1, fp )
110 fclose( fp )
120 /*
130 fp = fopen( dir$+"15パズル.x", "w" )
140 fseek( fp, &H1A24+64, 0 )
150 fwrite( code, 1, fp )
160 fwrite( code, 1, fp )
170 fclose( fp )
180 /*
190 fp = fopen( dir$+"パス名.x", "w" )
200 fseek( fp, &H116A+64, 0 )
210 fwrite( code, 1, fp )
220 fwrite( code, 1, fp )
230 fclose( fp )
```

図1　SXeyes.Xをデバッガで逆アセンブルしてみた例

```
-l 13773c 137778
  0013773C     move.l  $00137AF4,D0  ← まずはスタックサイズを取り出し
  00137742     cmp.l   #$00001000,D0 ← $1000と比較
  00137748     bcc.s   $00137750
  0013774A     move.l  #$00001000,D0 ← 小さければ$1000をセットして
  00137750     bclr.l  #$0000,D0
  00137754     move.l  D0,$00137AF4  ← 元の場所に書き込む
  0013775A     move.l  $00137AF8,D0  ← ヒープサイズを取り出し
  00137760     cmp.l   #$00002000,D0 ← 今度は$2000と比較
  00137766     bcc.s   $0013776E
  00137768     move.l  #$00002000,D0 ← 小さければ$2000をセットし
  0013776E     add.l   #$0000000D,D0
  00137774     bclr.l  #$0000,D0
  00137778     move.l  D0,$00137AF8  ← 元の場所へ書き込む
```

図2　2つの数値の実際の使われ方

```
-l 136fbe 136fdc
  00136FBE     move.l  #$00000400,D0 ← $400をセットし、
  00136FC4     add.l   $001372F4,D0  ← それに先の値を
  00136FCA     add.l   $001372F8,D0  ← 順次加えていく
  00136FD0     add.l   $00138018,D0  ← （こいつは意味不明）
  00136FD6     move.l  D0,-(A7)      ← サイズをスタックに積み
  00136FD8     pea     (A1)          ← ハンドルを積んで
  00136FDA     __MMSETHANDLESIZE     ← サイズを変更
  00136FDC     addq.l  #8,A7
```

C言語で使うグラフマネージャ基礎

GRAPHMANを使ってみよう

Izumi Daisuke 泉 大介

ウィンドウ上のグラフィック表示を一手に引き受けるのがグラフマンです。ここでは，レクタングル，リージョンといった独特の表現を使うグラフマンそのものの理解を目指して，ウィンドウとは切り離した解説を行ってみましょう。

SX-WINDOWでのプログラム作りがボチボチとスタートしました。先月の付録ディスク（DISK4）で提供されたドキュメントとリファレンスは，これまで闇の中でひたすら光明を求めていた私のような一般ユーザーにもアプリケーション作りの道を拓いてくれたのです。とはいっても，リファレンスもドキュメントも，わかる人のために書かれているといった感が強く，ウィンドウプログラミングの経験がない大方の皆さんは相変わらずの手探り状態が続いていることでしょう（私とて同じで，リセットスイッチを使ったのは，Z80のアセンブリ言語を使い始めたとき以来です）。

SX-WINDOW上で動くアプリケーションは，それぞれがひとつのタスクと見なされタスクマンの管理下に置かれるということは，先月の特集，そして今月の村田氏の記事でもおわかりいただけると思います。意地の悪い言い方をすれば，タスクマンに首根っこを押さえつけられてもがいているわけです。これまでのように，メモリは全部俺のもの，画面も全部俺のもの。スクロールが遅い？　ラスタスクロールしちゃえ！　といった自由奔放なプログラムは許されません。お行儀よくタスクマンに「なにかイベントは起きましたでしょうか」とお伺いをたて，画面上に同時に開いているであろうほかのウィンドウの迷惑にならないよう，そして一緒に動いているほかのタスクの迷惑にならないように，プログラムを作らなければならないのです。

それが作法というものさ，とドキュメントを読んでいた私の目は，グラフマンのドキュメントではたと止まります。「いろいろ揃えたねぇ。な〜んだ文字はプロポーショナルじゃないのか」と流し読みしていた私の目に，「グラフマンの初期化。グラフマンをSX-WINDOW以外で使用するためには……」という文が映ったのです。そういえば，GRSAMP.XはSX-シェルを起動せずに動いていたな。思い出しました。TSHELL.XなどはSXWDB.Xをまず起動し，それから動いていたのに，GRSAMP.Xはいきなり画面にグラフィックを表示しました。SXWDB.XやSXWIN.Xから起動したときには，COMMAND.XをSX-WINDOWから起動したときと同じようにSX-WINDOWを抜けてから動き始めました。ドキュメントから察するに，どうやらグラフィックライブラリとして使えるようなのです。

これならウィンドウプログラミングの経験も，パワーの有無も関係ありません。私にもなんとかなりそうです。IOCSコールのグラフィックルーチンは基本的なものばかりでしたが，こちらにはちょっと面白そうなものも入っていて遊べそうです。

まずは基礎知識の獲得から

ドキュメントによると，グラフマンを使うには，

1）　メモリマンを初期化
2）　グラフマンを初期化
3）　描画
4）　終了（特になにもしなくていい）

という手順でOKのようです。ただこれには但し書きがついていて，

「スーパーバイザのスタックは十分に用意すること。なかにはワークに10Kバイトほど使うものがある」

とあります。

さて，Cで作成したプログラムは通常ユーザーモードで動きます。これをスーパーバイザモードに切り替えるのはSUPERというDOSコール用の関数です。この関数を実行すると，それまでのユーザースタックがスーパーバイザスタックとして使用されるようになります。Cで作ったプログラムでは，通常（ユーザーが特に指定しない限り）ユーザースタックは64Kバイト確保されていますから，上の但し書きはクリアしたものと思っていいでしょう。こうしてスーパーバイザモードにしたあとはひたすら突っ走り，最後にユーザーモードに戻して終了するというアプローチで大丈夫です。では手順のほうをもう少し詳しく見ていきましょう。

●メモリマンを初期化

メモリマンはヒープを管理し，要求があればヒープからメモリを切り出して分けてくれます。mallocのSX-WINDOW版なわけですね。mallocと違うのは，要求された大きさのメモリを確保できなければ，内部でデータを詰め詰めにしてメモリを確保しようとしてくれることです。「ゲゲッ，確保したメモリのアドレスが変わってしまうのか」という心配はご無用。mallocで返されるのは「void *」ですが，メモリマンが返すのはいわば「void **」です。つまり，「void *」を入れたアドレス（こっちは固定）を返してくれるわけです。このあたりは先月村田氏が図解してくれましたが，あとでC言語的に少し解説する予定です。

メモリマンの初期化とは，「ヒープ領域はここからここまでね」とメモリマンに宣言することにほかなりません。SX-WINDOW（SXKERNEL，SXWDB，SXWIN）を使っているときにはシステムが勝手にやってくれますが，グラフマンを単独で使おうというときには自分でやらなければならないのです。これはmallocでメモリを確保し，それをメモリマン用のヒープとするのが簡単に実現できそうです。

●グラフマンの初期化

グラフマンは描画をgraph構造体を参照しながら行います。したがってここではgraph構造体の初期化も同時に行う必要があります。グラフマン自身の初期化はGMInitialize関数で，graph構造体の初期化はGMOpenGraph関数で行います。GMInitializeのほうは呼び出すだけ，GMOpenGraphのほうはgraph構造体を用意して，そのアドレスとテキスト画面にするかグラフィック画面も使うかを引数として渡すだけです。

●描画・終了

描画はこれからいくつでもサンプルが出てくるでしょうから，ここで取り上げなく

てもいいでしょう。終了処理はなにもしなくていいとのことなので，素直にそれを信じることにします。GRSAMP.Sでもなにもやっていないので，きっと大丈夫なのでしょう。

最初はGRSAMP.Sの真似っ子

なにもないところからいきなりプログラムを作り始めるのは大変です。まずは付録ディスクについてきたGRSAMP.Sを参考に，C言語で作り直してみたのがリスト1です。大域変数は大域変数のままに，変数名も変えずに作ってありますので（ただし場所はファイル先頭へ移しました）見比べてみてください。

●大域変数の宣言

最初に#defineしているのは，メモリマンに渡すメモリサイズです。GRSAMP.Sにならって128Kバイトをヒープとして渡しています。続く#ifdefはGCCユーザーへの配慮です。GCCでコンパイルするとmalloc用のヒープ領域は64Kバイトに設定されてしまい，XCのライブラリにあるmallocでは64K以上のメモリを確保できません。メモリマンには128Kバイト渡すつもりですから，ここでど～んと200Kバイトのヒープを取るように指示しています。巷に出回っているUNIXライブラリをご利用の方は削除してください。

続いてGRSAMP.Sにあった大域変数を宣言しています。t_graphはGMOpenGraph関数に渡すためのgraph構造体，scr_rectは画面全部を意味するレクタングル，r_rectとr_ovalはかどの丸い四角形（四円形と勝手に命名）を表現するレクタングルと楕円，そしてppatはペンのパターンです。ppatは四円形を描くときに指定される市松模様のペンデータです。残念ながらこの部分は問題があるようで，四円形の枠を市松模様にすることはできません。

●メモリマンの初期化

そしてプログラムはメインルーチンへとなだれ込みます。最初の予定どおりまずスーパーバイザモードに移行。次にメモリマンのヒープ用メモリをmallocしたら，メモリマンの初期化です。ヒープ領域の先頭アドレス，最終アドレス，一度に確保されるマスタポインタの数（GRSAMP.Sにならって100），エラー処理ルーチンのアドレス（別に指定しなくてもいいらしい），ブロックコンテンツの初期化フラグを引数にMMHeapInitを呼び出します。

●グラフマンの初期化

次はグラフマンの初期化です。GMInitializeで初期化し，t_graphをテキスト画面として初期化します。

●画面の塗り潰し

あとは絵を描いていくだけです。まずは背景色による画面の塗り潰しです。GMBackColor関数を使って背景色をセットします。ここでセットしているG_BLACKはSXDEF.Hで定義されていて，黒を指示します。続いてペンモードのセットです。ペンモードは，色の指定と描画ルールの2つの値を加えて指示します。色指定には，

1） G_FORE　：前景色
2） G_BACK　：背景色
3） G_PPAT　：ペンパターンで
4） G_EPAT　：拡張パターンで

の4つが，描画ルールには，

1） G_PSET　：指定色で
2） G_AND　：指定色と描画面のAND
3） G_OR　：同OR
4） G_XOR　：同XOR
5） G_NPSET　：指定色の反転
6） G_NAND　：指定色の反転と

リスト1　GRSAMP.SをC言語で

```
  1: #include <stdio.h>
  2: #include <stdlib.h>
  3: #include <sxlib.h>
  4:
  5: /* メモリマンに渡すメモリサイズを定義 */
  6:
  7: #define HeapSize     0x20000
  8:
  9: /* GNU Cではヒープが64Kに設定されるので、
 10:    それを200Kに変更する */
 11:
 12: #ifdef __GNUC__
 13:       asm( ".xdef _HEAP_SIZE" );
 14:       asm( "_HEAP_SIZE equ 204800");
 15: #endif
 16:
 17: /* プロトタイプ宣言 */
 18:
 19: int SUPER( int );
 20:
 21: /* graph用バッファ */
 22:
 23: graph    t_graph;
 24:
 25: /* 画面全体を示すrectangle */
 26:
 27: rect     scr_rect = { 0, 0, 768, 512 };
 28:
 29: /* round rectangleに外接するrectangleと
 30:    round rectangleのかどのoval */
 31:
 32: rect     r_rect = { 400, 20, 600, 120 };
 33: point_t r_oval = 0x000F000F;
 34:
 35: /* penのパターン (C == %1100, 3 == %0011) */
 36:
 37: unsigned short  ppat[ 16 ] = {
 38:       0xCCCC, 0xCCCC, 0x3333, 0x3333,
 39:       0xCCCC, 0xCCCC, 0x3333, 0x3333,
 40:       0xCCCC, 0xCCCC, 0x3333, 0x3333,
 41:       0xCCCC, 0xCCCC, 0x3333, 0x3333
 42: };
 43:
 44: /*
 45:  * メインルーチン (これしかない)
 46:  */
 47: void main()
 48: {
 49:       int      ssp;          /* ssp保存用 */
 50:       char     *memstart;   /* heapのスタートアドレス */
 51:       Heap     *heap;        /* heapの確保判定用 */
 52:
 53: /* スーパーバイザ領域に移動 */
 54:       ssp = SUPER( 0 );
 55:
 56: /* mallocによってヒープ用のメモリを確保 */
 57:       if (( memstart = (char *)malloc( HeapSize )) == NULL ) {
 58:             fprintf( stderr, "no enough memory for heap¥n" );
 59:             exit( 1 );
 60:       }
 61:
 62: /* メモリマンの初期化 */
 63:       heap = MMHeapInit( memstart,
 64:                          (char *)(memstart + 0x20000 ),
 65:                          100, NULL, 0 );
 66:       if ( heap == 0 ) {
 67:             fprintf( stderr, "mm : can't create heap¥n" );
 68:             exit( 1 );
 69:       }
 70:
 71: /* 初期設定 */
 72:       GMInitialize();                      /* グラフマンの初期化 */
 73:       GMOpenGraph( G_TXT, &t_graph ); /* graph構造体の初期化 */
 74:
 75: /* 画面の塗り潰し (背景色を使用) */
 76:       GMBackColor( G_BLACK );             /* 背景色設定 */
 77:       GMPenMode( G_BACK + G_PSET );      /* ペンモード設定 */
 78:       GMFillRect( &scr_rect );            /* 画面の塗り潰し */
 79:
 80: /* 線の描画 (前景色を使用) */
 81:       GMForeColor( G_WHITE );             /* 前景色設定 */
 82:       GMPenMode( G_FORE + G_PSET );      /* ペンモード設定 */
 83:       GMPenSize( 0x00010001 );            /* ペンサイズ設定 */
 84:       GMMove( 0x00100010 );          /* (0x10,0x10)にペンを移動 */
 85:       GMLine( 0x01000100 );          /* (0x100,0x100)まで線を引く */
 86:
 87: /* 四円形の枠を描画 (ペンパターンを使用) */
 88:       GMForeColor( G_WHITE );             /* 前景色設定 */
 89:       GMBackColor( G_DGRAY );             /* 背景色設定 */
 90:       GMPenMode( G_PPAT + G_PSET );      /* ペンモード設定 */
 91:       GMPenPat( ppat );                   /* ペンパターン設定 */
 92:       GMPenSize( 0x00040004 );            /* ペンサイズ設定 */
 93:       GMFrameRRect( &r_rect, r_oval ); /* 四円形を描く */
 94:
 95: /* <ASCIIZ> 文字列を描画 */
 96:       GMForeColor( G_BLACK );             /* 前景色設定 */
 97:       GMBackColor( G_WHITE );             /* 背景色設定 */
 98:       GMFontKind( G_ROM24 );              /* フォント設定 */
 99:       GMFontFace( G_ITALIC + G_OLINE ); /* 字体設定 */
100:       GMFontMode( G_PSET );               /* 表示方法設定 */
101:       GMFontSize( 0x00300030 );           /* 文字サイズ */
102:       GMMove( 0x00400120 );          /* (0x40,0x120)にペンを移動 */
103:       GMDrawStrZ( "abcdefg" );            /* 文字表示 */
104:
105: /* ユーザモード復帰 */
106:       SUPER( ssp );
107: }
```

図1　レクタングルの組成とscr_rec

レクタングル構造体

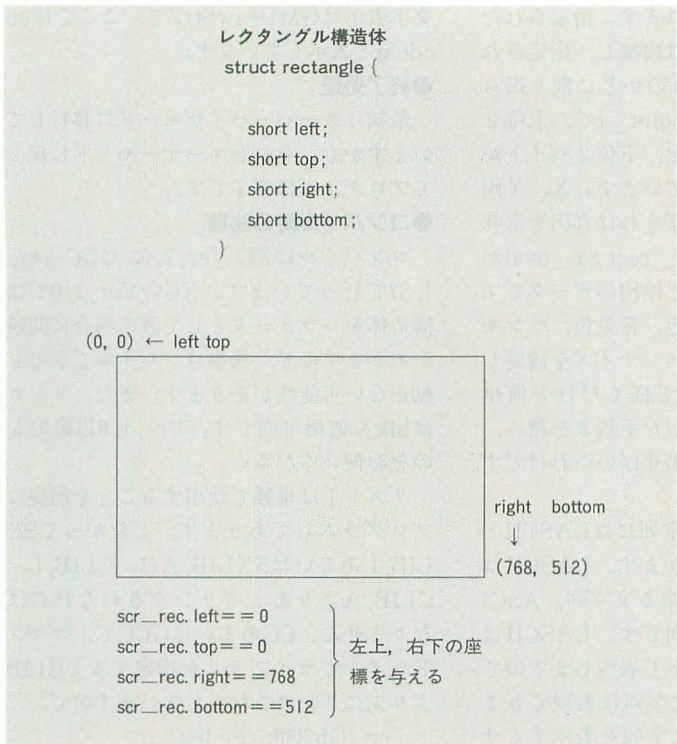
```
struct rectangle {

    short left;
    short top;
    short right;
    short bottom;
}
```

(0, 0) ← left top

right　bottom
↓
(768, 512)

```
scr_rec. left = = 0
scr_rec. top = = 0
scr_rec. right = = 768
scr_rec. bottom = = 512
```
左上，右下の座標を与える

図2　楕円と四円形の組成

　　　　　描画面のAND
7）　G_NOR　　：同OR
8）　G_NXOR　 ：同XOR
　以下65536色モード（未サポート）用
9）　G_ADD　　：2色を加える
10）　G_ADDLIM ：同上（最大値付）
11）　G_SUB　　：描画面から
　　　　　　　　指定色を引く
12）　G_SUBLIM ：同上（最小値付）
13）　G_SELMAX：大きい色を選ぶ
14）　G_SELMIN ：小さい色を選ぶ
15）　G_BLEND　：混ぜ合わせる
の15がSXDEF.Hで定義されています。

ここで使っているのはG_BACK+G_PSETですから，背景色での描画となります。ここでひとつ注意があります。先月のDISK4に入っていたSXDEF.Hでは，4つの色指定が，

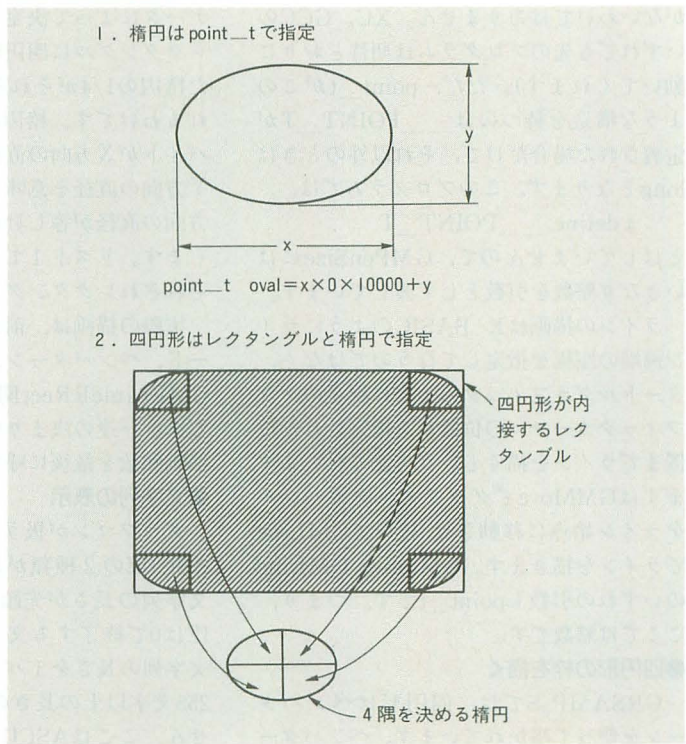
```
#define G_FORE $000
```

のように定義されています。このままではコンパイルできませんので，

```
#define G_FORE 0x000
```

と，$を0xに変更してください。ペンモードを設定したら，GMFillRectでレクタングルの塗り潰しを行います。

●ラインの描画

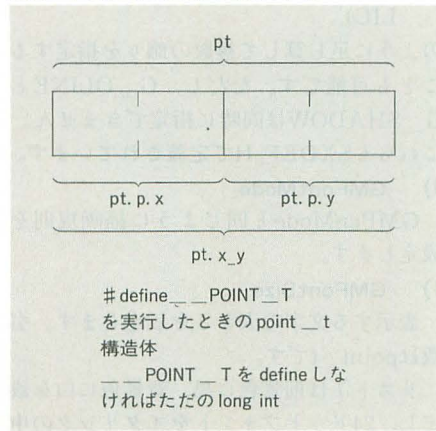
次は前景色を使ったラインの描画です。前景の色をGMForeColorでセットし，先ほどと同じようにGMPenModeでペンモードをセットします。今度は前景色を使いますからモードはG_FORE+G_PSETです。そしてGMPenSizeでペンの大きさを指定します。引数は0x00010001と16進8桁になっています。前半の4桁がX方向の大きさ，後半の4桁がY方向の大きさです。

ここでGMPenSizeの引数について補足しておきましょう。SXLIB.HにあるGMPenSizeのプロトタイプ宣言を見てみると，

```
int GMPenSize(point_t.);
```

となっています。SXDEF.Hによるとpoint_tは，

```
union {
    point p;
    long x_y;
}
```

だと定義されています。そしてpointは，

```
struct    {
    short x;
    short y;
}
```

と定義されています。つまりpoint_tは，long（4バイトの整数）としても，short+short（2バイトの整数2つ）としても扱うことができるようになっているのです。

座標を，

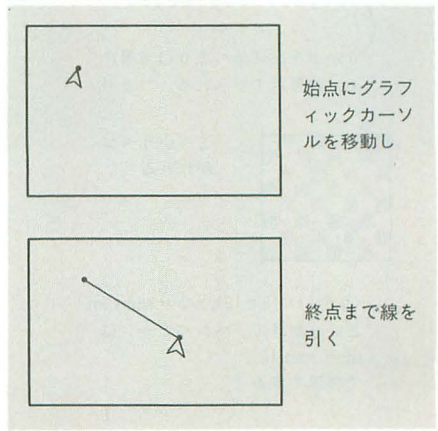
```
point_t pt;
pt.x_y = 0x00100100;
```

と一気に指定してしまうことも可能ですし，

```
pt.p.x += 10;
```

のように，X座標だけをあとから変更することも可能です（もっともこれは，short+shortとlongがコンパイラによって同じサイズに割り当てられる場合だけです。現行のたいていのコンパイラで先のプログラムは正しく処理されることが期待できます

図3　point_tを2通りに使う

図4　lineの描画方法

が，場合によってはうまくいかない可能性がないわけではありません。XC，GCCのいずれでも先のプログラムは期待どおりに動いてくれます)。ただ，point_tがこのような構造を持つのは___POINT_Tが定義された場合だけで，それ以外のときはlongとなります。このプログラムでは，

#define ___POINT_T

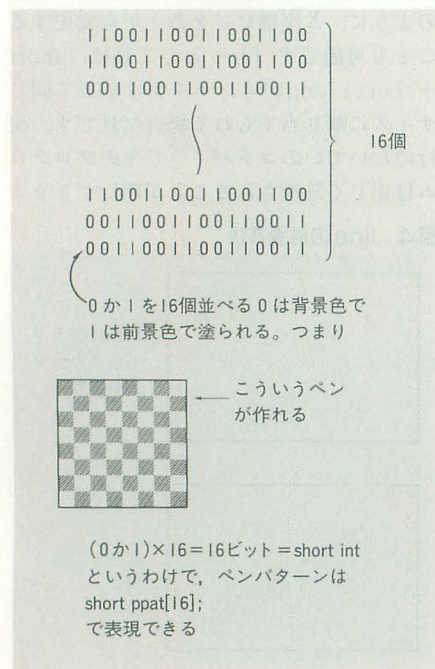
とはしていませんので，GMPenSizeにはいきなり整数を引数として渡しています。

ラインの描画はX-BASICのようにライン両端の座標を指定して行うのではなく，タートルグラフィックのように現在のグラフィックカーソルの位置から指定された位置までラインを描くという方法で行います。まずはGMMoveでグラフィックカーソルをライン始点に移動させ，続いてGMLineでラインを描きます。GMMove，GMLineのいずれの引数もpoint_tです。つまり，ここでは整数です。

●四円形の枠を描く

GRSAMP.Sでは，四円形はペンパターンを使って描かれています。ペンパターンはX方向16ドットのデータを16個並べたもので表現されます。1ドット分のデータは0か1で指示され，0ならそのドットを背景色に，1ならそのドットを前景色にするという意味になります。つまりX方向のデータは16ビット（2バイト：short int）で表現され，それが16個集まってパターンデータを形成しているわけです。リスト1ではppat配列がこのペンデータです。

四円形は，その大きさを決めるレクタン

図5　ペンパターンの組成

グルと，かどの丸みを決める楕円の2つのデータによって決定されます。指定されたレクタングルに四円形は内接し，指定された楕円の1/4がそれぞれのかどに割り振られるわけです。楕円はpoint_tで，上位2バイトがX方向の直径を，下位2バイトがY方向の直径を意味しています。X，Y両方向の直径が等しければそれは真円を意味します。リスト1ではr_rectとr_ovalがそれぞれレクタングルと楕円のデータです。

実際の描画は，前景色，背景色，ペンモード，ペンパターン，ペンサイズを設定し，GMFrameRRect関数で描くだけと簡単です。一連の決まり切った手続きを踏み，描画関数を最後に呼び出せばいいわけです。

●文字列の表示

グラフマンが扱う文字列にはLASCIIとASCIIZの2種類があります。LASCIIは文字列の長さが先頭に来る文字列。ASCIIZは0で終了する文字列です。LASCIIは文字列の長さを1バイトで表現しますので，255文字以上の長さの文字列は表現できません。ここはASCIIZ文字列を表示するサンプルになっています。

文字列表示ではいくつかの情報設定が必要となります。以下にこれらの情報を設定する関数を挙げます。

1)　GMFontKind

表示するフォントを設定します。フォントはG_ROM12，G_ROM16，G_ROM24の3つのフォントを選ぶことができます。いわずと知れたROM内フォントとIOCSで作られる12ドットフォントで，G_ROM12などはSXDEF.Hで定義されています。

2)　GMFontFace

文字飾りを設定します。太字（G_BOLD），イタリック（G_ITALIC），下線付（G_ULINE），中抜文字（G_OLINE），影付き中抜文字（G_SHADOW）の5種類があり，

GMFontFace(G_BOLD＋G_ITALIC)

のように足し算して複数の飾りを指定することも可能です。ただし，G_OLINEとG_SHADOWは同時に指定できません。これらもSXDEF.Hで定義されています。

3)　GMFontMode

GMPenModeと同じように描画規則を設定します。

4)　GMFontSize

表示する文字の大きさを設定します。引数はpoint_tです。

リスト1は前景色に黒，背景色に白を設定し，24ドットフォントをイタリックの中抜きで48×48ドットに拡大して表示します。文字表示はGMDrawStrZで，ここではabcdefgと表示しています。

●終了処理

最初にスーパーバイザモードに移行していますから，それをユーザーモードに戻してプログラムは終了です。

●コンパイル時の注意

コンパイルはXC Ver.2.0，GCC Ver.1.37で行っています。XCのVer.1.0では構造体をパラメータとして渡す場合に問題がありますので，無事コンパイルできても動かない可能性があります。また，リンカはhlkも使用可能です。Ver.1.6以降のものをお使いください。

リスト1は単独で使用することを前提にプログラムしてあります。したがってSXLIB.LあるいはSXLIB.Aは，CLIB.L，CLIB.Aよりあとでリンクされなければなりません。CCあるいはGCCで，コマンドラインでライブラリを指定するとCLIBより先にリンクされてしまいますので，

```
cc /Gh200k /Fc list1.c
gcc -c -fno-defer-pop list1.c
```

としてlist1.oをまず作り，それから，

```
lk list1.o clib.l …… sxlib.l
```

のようにリンク作業を行う必要があります。gccの-fno-defer-popオプションは，関数呼び出し時にスタックに積んだ引数を，呼び出しが終了するたびに取り除くようにする指定です。通常はいくつかまとめて取り除くようになっていますので，SUPERを呼び出したときにバスエラーを起こしてしまいます。忘れないようにしてください。

リンク時にライブラリを列挙するのは面倒なものなので，リスト3のようなメイクファイルを作っておくと便利でしょう。これはXC Ver.2.0に付属してきたmake.x用のメイクファイルです。これと同じものをmakefileというファイル名で作成し，カレントディレクトリに置いておけば，

```
make
```

と入力するだけでコンパイルからリンクまで自動的に作業が行われます。他のプログラムを作成するときには，「prog＝」の後ろのファイル名を変更するだけでOKです。ライブラリの存在するディレクトリ名は，皆さんの環境に合わせて書き換えて使ってください。

なお，gccを使っている方には悲しいお知らせがあります。メモリ2Mバイトではこのプログラムはコンパイルできません。どうしてもgccでコンパイルしたい方は，リスト2のようなヘッダファイルを自分で

作り，

　#include <sxlib.h>

のところを，

　#include <list1.h>

のように書き換えてコンパイルしてください。list1.hはlist1.c用のヘッダファイルです。これを使うとフリーエリア1.3Mバイトでコンパイル可能となります。

表示される色がおかしい？

リスト1をコンパイル・実行してみると，色が変なことに気がつきます。背景はG_BLACKで塗り潰しているはずなのに白ですし，四円形は黄色混じりで表示されます。この理由は，テキストのパレットが標準の4色のままになっているからです。SX-WINDOWでは8色使用することができ，SXDEF.Hで定義されているB_BLACKなどはこの8色モード用の色なのです。GRSAMP.SにはSX-WINDOW用のカラーに設定するGMInitPaletの呼び出しと，テキスト4プレーン全部に描画できるようにするGMAPageがありますが，これらはコメントにして殺されています。

これを考慮してリスト1を作り直したのがリスト4です。プログラムをすっきりさせるため，初期・終了処理をmainで行い，graph構造体を使用する描画に関係した処理はdisplay関数にまとめました。

初期処理として付け加えたのは，カーソルを消去することとパレットの初期化を行うことです。このあとdiaplay関数を呼び出して描画を行ったあと，終了処理に入ります。画面の最下行でカーソルを点滅させてキー入力を待ち，テキスト画面をクリアしてパレットをHuman用に再初期化。ファンクションキーの再表示を経てユーザーモードに戻り終了します。

display関数ではGMAPage関数を呼び出して，テキスト4プレーンすべてに描画できるようにしたほかはリスト1と同じで

リスト2　小さなヘッダファイル

```
  1: typedef struct TBM {
  2:         int             page;           /* 1ページのバイト数 */
  3:         unsigned short  aPage;          /* 描画画面の指定 */
  4: } TBM;
  5:
  6: typedef struct rect {                   /* rectangle */
  7:         short   left;                   /* 左端の座標 */
  8:         short   top;                    /* 上端の座標 */
  9:         short   right;                  /* 右端の座標 */
 10:         short   bottom;                 /* 下端の座標 */
 11: } rect;
 12:
 13: typedef struct bitmap {
 14:         short   bmKind;                 /* 画面の種類 */
 15:         rect    bmRect;                 /* 画面の大きさ */
 16:         int     base;                   /* ベースアドレス */
 17:         short   line;                   /* 横1ラインのバイト数 */
 18:         union {
 19:                 unsigned short  bRatio; /* 色の混ぜ合わせの比率 */
 20:                 TBM     tbm;
 21:         } opt;
 22: } bitmap;
 23:
 24: typedef struct region {
 25:         int     size;                   /* 全体のバイト数 */
 26:         rect    bounds;                 /* 囲む四角形 */
 27: } region;
 28:
 29: typedef long    point_t;
 30:
 31: typedef struct graph {
 32:         bitmap          *bmap;          /* bitmap へのポインタ */
 33:         rect            grRect;         /* 画面サイズ */
 34:         int             *procs;         /* 描画ルーチンテーブル */
 35:         region          **visible;      /* 描画可能領域のリージョン */
 36:         region          **clipping;     /* 描画可能領域のリージョン */
 37:         short           drawLvl;        /* 描画レベル */
 38:         unsigned short  penMode;        /* 描画モード */
 39:         point_t         penLoc;         /* 描画位置 */
 40:         point_t         penSize;        /* ペンの太さ */
 41:         short           *penPat;        /* ペンパターン */
 42:         short           *exPat;         /* 拡張パターン */
 43:         short           workKind;       /* システムで使用 */
 44:         void            **workHdl;      /* システムで使用 */
 45:         unsigned short  fgColor;        /* 描画色 */
 46:         unsigned short  bgColor;        /* 背景色 */
 47:         short           fontKind;       /* フォントの種類 */
 48:         unsigned short  fontFace;       /* 飾り文字の指定 */
 49:         unsigned short  fontMode;       /* 描画モード */
 50:         point_t         fontSize;       /* 文字の大きさ */
 51: } graph;
 52:
 53: typedef struct Heap {
 54: } Heap;
 55:
 56: Heap    *MMHeapInit(char *, char *, int, int (*)(), int);
 57: void    GMInitialize(void);
 58: int     GMOpenGraph(int, graph *);
 59: int     GMBackColor(int);
 60: int     GMPenMode(int);
 61: int     GMFillRect(rect *);
 62: int     GMForeColor(int);
 63: int     GMPenSize(point_t);
 64: void    GMMove(point_t);
 65: int     GMLine(point_t);
 66: unsigned short  *GMPenPat(unsigned short *);
 67: int     GMFrameRRect(rect *, point_t);
 68: int     GMFontKind(int);
 69: int     GMFontFace(int);
 70: int     GMFontMode(int);
 71: int     GMFontSize(point_t);
 72: int     GMDrawStrZ(char *);
 73:
 74: /*
 75:  *      <screen kind>
 76:  */
 77: #define G_TXT           0
 78: #define G_GRP           1
 79: /*
 80:  *      <draw mode>
 81:  */
 82: #define G_PSET          0
 83: #define G_AND           1
 84: #define G_OR            2
 85: #define G_XOR           3
 86: #define G_NPSET         4
 87: #define G_NAND          5
 88: #define G_NOR           6
 89: #define G_NXOR          7
 90: #define G_ADD           8
 91: #define G_ADDLIM        9
 92: #define G_SUB           10
 93: #define G_SUBLIM        11
 94: #define G_SELMAX        12
 95: #define G_SELMIN        13
 96: #define G_BLEND         14
 97: /*
 98:  *      pen mode
 99:  */
100: #define G_FORE          0x000
101: #define G_BACK          0x100
102: #define G_PPAT          0x200
103: #define G_EPAT          0x300
104: /*
105:  *      <font kind>
106:  */
107: #define G_ROM12         0
108: #define G_ROM16         1
109: #define G_ROM24         2
110: /*
111:  *      <font face>
112:  */
113: #define G_BOLD          1
114: #define G_ITALIC        2
115: #define G_ULINE         4
116: #define G_OLINE         8
117: #define G_SHADOW        16
118: /*
119:  *      <font face> ビット番号
120:  */
121: #define GB_BOLD         0
122: #define GB_ITALIC       1
123: #define GB_ULINE        2
124: #define GB_OLINE        3
125: #define GB_SHADOW       4
126: /*
127:  *      text color
128:  */
129: #define G_THRU          0       /* 透明 */
130: #define G_WHITE         8       /* 白 */
131: #define G_LGRAY         9       /* 明るいグレー */
132: #define G_DGRAY         10      /* 暗いグレー */
133: #define G_BLACK         11      /* 黒 */
134: #define G_YELLOW        12      /* 黄色 */
135: #define G_RED           13      /* 赤 */
136: #define G_GREEN         14      /* 緑 */
137: #define G_BLUE          15      /* 青 */
```

リスト3　メイクファイルの例

```
  1: prog = list1
  2: obj = $(prog).o
  3: lib =   ¥
  4:         ¥lib¥clib.l ¥lib¥floatfnc.l¥
  5:         ¥lib¥doslib.l¥
  6:         ¥lib¥iocslib.l¥
  7:         ¥lib¥sxlib.l
  8:
  9: $(prog).x: $(obj)
 10:         lk $(obj) $(lib)
 11:
 12: %.o: %.c
 13:         cc /Gh200k /Fc $<
```

す。GMAPageはgraph構造体内のデータを変更しますので，まずGMOpenGraphでgraph構造体を初期化してから指定します。

これで以後の作業はdisplay関数内を書き換えるだけでOKとなりました。グラフマンを使用するプログラムの雛形が完成したわけです。

リージョンいろいろ

ドキュメントには，レクタングルや楕円，四円形などでは表現できない複雑な形を表現する手段として，リージョンが紹介されています。リージョン（region）とは領域という意味で，表示したい形に外接するレクタングルと表示のON/OFFを表現するラインデータから成っています。

ラインデータは，

y1 xs11 xe11 xs12 …… 0x7FFF
y2 xs21 xe21 xs22 …… 0x7FFF
……
0x7FFF

という形をしています。これは指定されたY座標yの，X座標xs～xeの間がリージョンに含まれていなければ含め，含まれているならば外すことを指示するデータ列です。設定した状態は，続くラインデータによって変更されるまで有効となります。ラインデータの最後は，先頭データであるY座標を0x7FFFにして指示するようになっています。

ドキュメントには例として，四角形が斜めに2つつながったような形などがありますね。今度はこれをC言語でやってみることにしましょう。

●リージョンを自分で設定する

ドキュメントにはアセンブラでプログラムを作成するときのリージョンの表記方法が付いていますが，C言語での方法は書いてありません。ラインデータがいくつつながるかわからない以上，アセンブラを使うときのように簡単には表記できないのです。実際SXDEF.Hにあるregion構造体の定義では，ラインデータのないレクトリージョンしか扱えないようになっています。

解決策のその1は，自分でmyRegionなどの名前で必要なラインデータが入る大きさの構造体を定義し，初期値代入でデータをセットしてしまうことです。ただこれは，すべてのregion用に構造体を別途用意して，しかも大域変数とすることを考えるとなかなかにみっともないものです。かといって，display関数の局所変数として

リスト4　SX-WINDOWのパレットで描画する

```
  1: #include <stdio.h>
  2: #include <stdlib.h>
  3: #include <sxlib.h>
  4:
  5: /* メモリマンに渡すメモリサイズを定義 */
  6:
  7: #define HeapSize     0x20000
  8:
  9: /* GNU Cではヒープが64Kに設定されるので、
 10:    それを200Kに変更する */
 11:
 12: #ifdef __GNUC__
 13:     asm( ".xdef _HEAP_SIZE" );
 14:     asm( "_HEAP_SIZE equ 204800");
 15: #endif
 16:
 17: /* プロトタイプ宣言 */
 18:
 19: int  SUPER( int );
 20: int  C_CUROFF(void);
 21: int  C_CURON(void);
 22: int  C_LOCATE(int, int);
 23: int  C_WIDTH(int);
 24: int  C_FNKMOD(int);
 25: void TXFILL( unsigned short * );
 26:
 27: void display( void );
 28:
 29: unsigned short clrdat[] = { /* 画面消去用 */
 30:     2,          /* テキストプレーン3の */
 31:     0, 0,       /* ( 0, 0 )から */
 32:     768, 512,   /* 横768、縦512ドットを */
 33:     0           /* 0で埋める */
 34: };
 35:
 36: /* GRAPHMAN用変数 */
 37:
 38: graph   t_graph;             /* graph構造体 */
 39:
 40: rect    scr_rect = {         /* 全画面 */
 41:     0, 0, 768, 512
 42: };
 43: `
 44: rect    r_rect = {       /* 四円体の外接レクタングル */
 45:     400, 20, 600, 120
 46: };
 47: point_t r_oval = 0x000F000F;    /* 4すみの楕円 */
 48:
 49: unsigned short  ppat[ 16 ] = {  /* ペンパターン */
 50:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 51:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 52:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 53:     0xCCCC, 0xCCCC, 0x3333, 0x3333
 54: };
 55:
 56: /*
 57:  *    メインルーチン
 58:  *      初期・終了処理と描画関数呼び出し
 59:  */
 60: void main()
 61: {
 62:     int ssp;                            /* ssp保存用変数 */
 63:     char    *memstart;
 64:     Heap    *heap;
 65:
 66:     /*
 67:      *     初期処理
 68:      */
 69:     ssp = SUPER( 0 );        /* スーパーバイザモードへ移行 */
 70:     if (( memstart = (char *)malloc( HeapSize )) == NULL ) {
 71:         fprintf( stderr, "no enough memory for heap¥n" );
 72:         exit( 1 );            /* ヒープ用メモリがなければ終了 */
 73:     }
 74:     heap = MMHeapInit( memstart,    /* ヒープの初期化 */
 75:                        (char *)(memstart + 0x20000 ),
 76:                        100, NULL, 0 );
 77:     if ( heap == 0 ) {
 78:         fprintf( stderr, "mm : can't create heap¥n" );
 79:         exit( 1 );
 80:     }
 81:     C_CUROFF();                      /* カーソルOFF */
 82:     GMInitialize();                  /* グラフマンの初期化 */
 83:     GMInitPalet();                   /* パレットの初期化 */
 84:
 85:     /*
 86:      *     メイン本体
 87:      */
 88:     display();
 89:
 90:     /*
 91:      *     終了処理
 92:      */
 93:     C_CURON();                       /* カーソルON */
 94:     C_LOCATE( 0, 30 );               /* 画面の一番下で */
 95:     getch();                         /* キー入力待ち */
 96:     TXFILL( clrdat );                /* テキストプレーン3消去 */
 97:     clrdat[ 0 ] = 3;                 /* テキストプレーン4を */
 98:     TXFILL( clrdat );                /* 消去 */
 99:     C_WIDTH( 0 );                    /* 768×512モード */
100:     C_FNKMOD( 0 );                   /* ファンクションキー表示 */
101:
102:     SUPER( ssp );                    /* ユーザモードへ */
103: }
104:
105: void display( void )
106: {
107: /* 初期設定 */
108:     GMOpenGraph( G_TXT, &t_graph ); /* graphをオープン */
109:     GMAPage( 15 );                      /* 全ページに描画 */
110:
111: /* 画面の塗り潰し(背景色を使用)*/
112:     GMBackColor( G_BLACK );             /* 背景色設定 */
113:     GMPenMode( G_BACK + G_PSET );       /* ペンモード設定 */
114:     GMFillRect( &scr_rect );            /* 画面の塗り潰し */
115:
116: /* 線の描画(前景色を使用)*/
117:     GMForeColor( G_WHITE );             /* 前景色設定 */
118:     GMPenMode( G_FORE + G_PSET );       /* ペンモード設定 */
119:     GMPenSize( 0x00010001 );            /* ペンサイズ設定 */
120:     GMMove( 0x00100010 );        /* (0x10,0x10)にペンを移動 */
121:     GMLine( 0x01000100 );        /* (0x100,0x100)まで線を引く */
122:
123: /* 四円形の枠を描画(ペンパターンを使用)*/
124:     GMForeColor( G_WHITE );             /* 前景色設定 */
125:     GMBackColor( G_DGRAY );             /* 背景色設定 */
126:     GMPenMode( G_PPAT + G_PSET );       /* ペンモード設定 */
127:     GMPenPat( ppat );                   /* ペンパターン設定 */
128:     GMPenSize( 0x00040004 );            /* ペンサイズ設定 */
129:     GMFrameRRect( &r_rect, r_oval ); /* 四円形を描く */
130:
131: /* <ASCIIZ> 文字列を描画 */
132:     GMForeColor( G_BLACK );             /* 前景色設定 */
133:     GMBackColor( G_WHITE );             /* 背景色設定 */
134:     GMFontKind( G_ROM24 );              /* フォント設定 */
135:     GMFontFace( G_ITALIC + G_OLINE ); /* 字体設定 */
136:     GMFontMode( G_PSET );               /* 表示方法設定 */
137:     GMFontSize( 0x00300030 );           /* 文字サイズ */
138:     GMMove( 0x00400120 );        /* (0x40,0x120)にペンを移動 */
139:     GMDrawStrZ( "abcdefg" );            /* 文字表示 */
140: }
```

宣言するのはスタックを食い潰してしまってうまくありません。それならというので，使う度に {} でブロックを作るというアイデアは捨ててください。画面に表示されているリージョンの本体がブロック終了と同時に破棄されて残らないというのは問題があるでしょう。

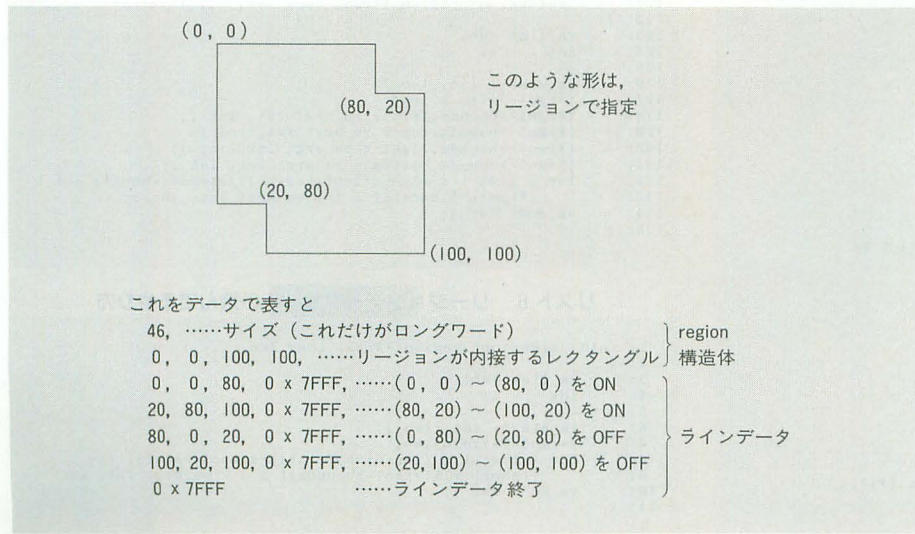
●リージョン設定の方針

結局ここでは，汎用性のある関数をひとつ作り，これを呼び出すことでデータのセットを行うことにしました。XC Ver.2.0ではANSI仕様に沿った方向でバージョンアップが図られましたが，このとき不定個の引数を取る関数を定義する方法もサポートされています。これを利用すればラインデータがいくつあっても大丈夫です。リージョンにデータをセットするこの関数をsetRegionと命名しておきましょう。

リージョンのデータを収める領域は，メモリマンにヒープを分けてもらって作ることにします。リージョンは，

```
struct {
    int   size;
    rect  bounds;
    short linedata [17];
}
```

のような形をしています。linedataを除いたものはregion構造体としてSXDEF.Hで定義されていますから，メモリマンに要求するヒープのサイズは，「sizeof (region) +sizeof (short) ×17」ということになります。もちろんラインデータの数が多ければ，17倍するところは変更する必要があります。

MMChHdlNew (sizeof……) でメモリマンにヒープを要求すると，メモリマンは要求されたサイズのメモリをヒープ領域から切り出し，確保したメモリの先頭アドレスを内部の固定領域に収めたあと，その固定領域のアドレスを返してくれます。返されたアドレスをadrsだとすると，*adrsにヒープから切り出されたメモリの先頭アドレスが入っているわけです。実際に確保されたメモリを操作するためには，さらにその先，つまり**adrsを参照しなければなりません。メモリマンが「void **」を返すと最初に説明したのはこのことです。メモリマンが返す内部の固定領域のアドレスのことをハンドルといいます。

受け取る変数のほうもこれに合わせて用意します。リージョン分のメモリを要求したのですから，

```
region **rgn;
rgn = MMChHdlNew(sizeof……);
```

とすればOKです。え？　要求したサイズはregion構造体のサイズより大きいのではないか，ですって。そのとおり。でもCコンパイラはそんな小さなことをウジウジ言うほど了見が狭くはありません。

●リージョン設定の実際

実際にリージョンを設定するには，SXDEF.Hのregion構造体の定義に若干の修正を加えます。

```
typedef struct region {
    int    size;
    rect   bounds;
    short  linedata [0];
} region;
```

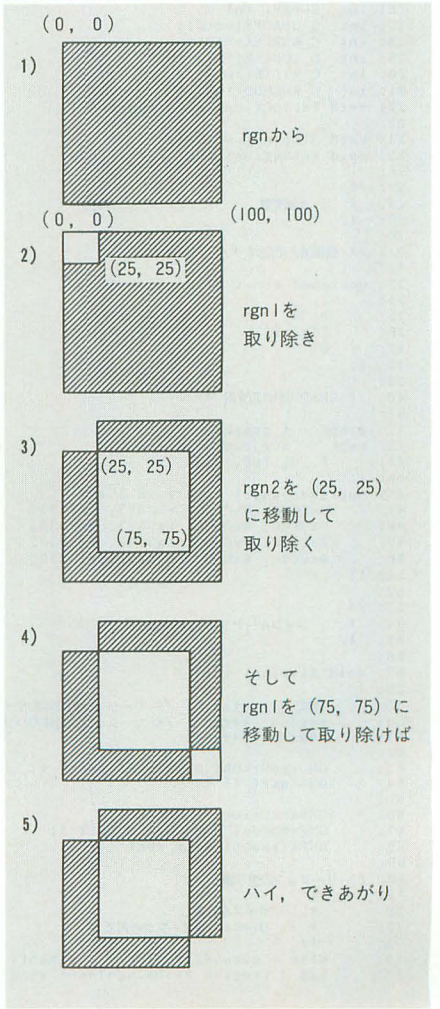
のように，linedata [0] という行を追加してください。配列サイズは0ですから，sizeof (region) は以前と変わりません。ただし，linedataという名前の配列が構造体のメンバに加わりましたので，先の例のように17個のshortを入れる領域を用意してやれば，linedata [0] ~linedata [16] でそれぞれの要素を扱えるようになったのです。setRegion関数の引数として渡されたラインデータを1つずつ取り出し，linedata配列にセットしていけばリージョンにデータをセットすることができます。このlinedataメンバはぜひともregion構造体に加えていただきたいのですが，シャープさん，いかがですか。

●リージョンを表示する

ではリスト5です。インクルードファイルにXC Ver.2.0に付いてきたヘッダファイルstdarg.hを追加し，プロトタイプ宣言のところにsetRegion関数のプロトタイプを追加。メイン関数はリスト4のものを使います。そしてdisplay関数をリージョン表示用に書き換えました。不定個の引数を取る関数は，必ず取る引数の宣言のあとに「...」と書いて引数が続くことを示します。

display関数では「region **」を2つ宣言して，2つのリージョンを同時に表示することにしてあります。では「リージ

図6　リージョンの組成

図7　正攻法のリージョン作り

ョンの塗り潰し」とコメントしてあるところから見ていきましょう。まずはsize変数に，必要とするリージョンのサイズを計算します。ここではラインデータを17個取るリージョンを表示します。サイズが計算できたらメモリマンにメモリを要求し，返されたハンドルを「region ＊＊」にキャストして変数に収めます。

そして次がリージョンのデータセットです。リージョンのハンドル，リージョンのサイズ，リージョンが内接するレクタングル，ラインデータをパラメータとしてsetRegion関数を呼び出します。ドキュメントのリージョンのサンプルのところに挙げてあった，アセンブラでの記述法と同じようにデータを並べていけば，指定したリージョンのハンドルにデータがセットされます。続いてもうひとつのリージョンも同様にメモリを確保し，データをセットします。

色とペンパターンを設定したら最後の仕上げ，リージョンの塗り潰しです。GMFillRgn関数で最初のリージョンを塗り潰し，2つ目のリージョンはGMSlideRgn関数で横に少しずらして塗り潰します。

最後にsetRegion関数ですが，stdarg.hで定義されているva_start，va_arg，va_endの3つのマクロを使って不定個の引数に対応します。必ず取る引数の最後のものをva_startで指定したら，不定個の部分はva_argを実行するたびにひとつずつ取り出されていきます。これを利用してリージョンのサイズ，リージョンが内接するレクタングルにデータをセットし，続いてラインデータをセットしていきます。最後にva_endを実行してデータのセットは終了です。

このsetRegion関数は，最初から用意されたライブラリ関数のように使うと便利でしょう。ドキュメントにあるリージョンのサンプルを，自分で入力して表示してみてください。

●FURTHER STUDY

実はSXDEF.Hのリージョン構造体の宣言を変更しなくてもラインデータをセットすることができます。リージョンが内接

リスト5　リージョンの描画

```
  1: #include <stdio.h>
  2: #include <stdlib.h>
  3: #include <stdarg.h>
  4: #include <sxlib.h>
  5:
  6: #define HeapSize    0x20000
  7: #ifdef __GNUC__
  8:     asm( ".xdef _HEAP_SIZE" );
  9:     asm( "_HEAP_SIZE equ 204800");
 10: #endif
 11:
 12: /*
 13:  *    プロトタイプ宣言
 14:  */
 15:
 16: int  SUPER( int );
 17: int  C_CUROFF(void);
 18: int  C_CURON(void);
 19: int  C_LOCATE(int, int);
 20: int  C_WIDTH(int);
 21: int  C_FNKMOD(int);
 22: void TXFILL( unsigned short * );
 23:
 24: void display( void );
 25: void setRegion( region **, long, ... );
 26:
 27: /*
 28:  *    大域変数
 29:  */
 30:
 31: /* 画面消去用配列 */
 32:
 33: unsigned short clrdat[] = {
 34:     2,
 35:     0, 0,
 36:     768, 512,
 37:     0
 38: };
 39:
 40: /* GRAPHMAN用変数 */
 41:
 42: graph   t_graph;
 43: rect    scr_rect = {
 44:     0, 0, 768, 512
 45: };
 46: unsigned short  ppat[ 16 ] = {
 47:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 48:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 49:     0xCCCC, 0xCCCC, 0x3333, 0x3333,
 50:     0xCCCC, 0xCCCC, 0x3333, 0x3333
 51: };
 52:
 53: /*
 54:  *    メインルーチンはリスト4と同じ
 55:  */
 56:
 57: void display( void )
 58: {
 59:     int     size;   /* リージョン構造体のサイズ */
 60:     region  **rgn;  /* リージョン構造体のハンドルアドレス */
 61:     region  **rgn1;
 62:
 63:     GMOpenGraph( G_TXT, &t_graph );
 64:     GMAPage( 15 );
 65:
 66:     GMBackColor( G_BLACK );
 67:     GMPenMode( G_BACK + G_PSET );
 68:     GMFillRect( &scr_rect );
 69:
 70: /* リージョンの塗り潰し */
 71:     /*
 72:      *    サイズを計算し
 73:      *    リージョンをヒープから得る
 74:      */
 75:     size = sizeof( region ) + sizeof( short )*17;
 76:     rgn = (region **)MMChHdlNew( size );
 77:     /*
 78:      *    リージョンの設定
 79:      */
 80:     setRegion( rgn,
 81:                size,
 82:                10, 110, 40, 140,
 83:                110, 10, 30, 0x7FFF,
 84:                120, 30, 40, 0x7FFF,
 85:                130, 10, 20, 0x7FFF,
 86:                140, 20, 40, 0x7FFF,
 87:                0x7FFF );
 88:
 89:     size = sizeof( region ) + sizeof( short )*17;
 90:     rgn1 = (region **)MMChHdlNew( size );
 91:     setRegion( rgn1,
 92:                size,
 93:                10, 110, 40, 140,
 94:                110, 20, 30, 0x7FFF,
 95:                120, 10, 40, 0x7FFF,
 96:                130, 10, 40, 0x7FFF,
 97:                140, 20, 30, 0x7FFF,
 98:                0x7FFF );
 99:
100:     GMForeColor( G_GREEN );             /* 前景色設定 */
101:     GMBackColor( G_DGRAY );             /* 背景色設定 */
102:     GMPenMode( G_PPAT + G_PSET );       /* ペンモード設定 */
103:     GMPenPat( ppat );                   /* ペンパターン設定 */
104:     GMPenSize( 0x00040004 );            /* ペンサイズ設定 */
105:
106:     GMFillRgn( rgn );                   /* リージョン塗り潰し */
107:     GMSlideRgn( rgn1, 0x00300000 ); /* rgn1を平行移動 */
108:     GMFillRgn( rgn1 );                  /* rgn1を塗り潰し */
109: }
110:
111: /*
112:  *    リージョンの設定
113:  *    SXDEF.Hのregionの定義を
114:  *    typedef struct region {
115:  *      int   size;
116:  *      rect  bounds;
117:  *      short linedata[ 0 ];
118:  *    } region;
119:  *    に変更すること
120:  */
121: void setRegion( region **rgn, long len, ... )
122: {
123:     va_list lst;
124:     int     i;
125:
126:     va_start( lst, len );
127:     (*rgn)->size = len;
128:     (*rgn)->bounds.left = va_arg( lst, int );
129:     (*rgn)->bounds.top = va_arg( lst, int );
130:     (*rgn)->bounds.right = va_arg( lst, int );
131:     (*rgn)->bounds.bottom = va_arg( lst, int );
132:     for ( i=0; i < (len-sizeof(region))/sizeof(short); i++ )
133:         (*rgn)->linedata[ i ] = va_arg( lst, int );
134:     va_end( lst );
135: }
```

リスト6　リージョンデータセットの飛んでるやり方

```
 1: void setRegion( region **rgn, long len, ... )
 2: {
 3:     va_list lst;
 4:     int     i;
 5:
 6:     va_start( lst, len );
 7:     (*rgn)->size = len;
 8:     for ( i=0; i < (len-sizeof(int))/sizeof(short); i++ )
 9:         ((short *)&(*rgn)->bounds)[ i ] = va_arg( lst, int );
10:     va_end( lst );
11: }
```

するレクタングルは構造体ですが，それぞれのメンバはshortであることを考えると，

```
struct region {
    int size;
    short bounds[4];
}
```

と見なすことも可能なのです。

これを利用すると，setRegion関数はリスト6のように書き換えることができます。ただ，これではなにをやっているのかわかりづらくなってしまいます。やはりlinedataメンバはサポートしてほしいところです。

正統派のリージョン描画

setRegion関数が用意されていないこと，また，region構造体にlinedataメンバがないことを考え合わせると，どうもリージョンを自分でセットするというアプローチは禁止されているのではないかという印象を拭いきれません。グラフマンのリファレンスを眺めていると，空のリージョン（ヌルリージョン）を作る，リージョンとリージョンの共通部分を得てリージョンとする，などの関数がそれを裏づけるかのようです。

これは，将来リソースエディタがサポートされたときには，SX-WINDOW上でマウスをちょちょいといじって自由な形のリージョンを定義できるようにするからということなのかもしれません。早くリソースエディタとリソースコンパイラをサポートしてほしいものです。

それでは正統派の手続きでリージョンを作るとどうなるのかを見ていただきましょう。リスト7です。大域変数はリスト5，メインルーチンはリスト4と同じなのでここでは省略してあります。また，point_tを16進数で指定するのはやっかいなので，プログラムの最初でX_Yというマクロを定義してあります。

display関数を見てみましょう。ここではリージョンのハンドルを3つ用意しました。rgnは描画するリージョンを作成するため，rgn1，rgn2は途中の加工用です。最初にGMNewRgn関数で，新しいリージョンを作成します。ここで作られるのは大きさのないヌルリージョンです。具体的にはリージョンが内接するレクタングルが，(0, 0)-(0, 0)となっています。そこでリージョンが内接するレクタングルの右下の座標を変更し，rgnは(0, 0)-(100, 100)に，rgn1は(0, 0)-(25, 25)に，rgn2は(0, 0)-(50, 50)にします。これで3つのヌルリージョンは，長方形の領域を表すレクトリージョンになりました。

まずはrgnとrgn1から重なった部分をGMXorRgnで取り除いたリージョンを作成し，それをrgnにセットします。つまりrgnは左上の25×25ドットが欠けた領域になるわけです。続いてGMMoveRgnでrgn2を(25, 25)に移動し，再びrgnとの共通部分を取り除きます。これでrgnは左上の25×25ドット，真ん中の50×50ドットが欠けた領域になりました。最後にrgn1を(75, 75)に移動して，rgnとの共通部分を取り除きます。こうして右下の25×25ドットも欠けた領域ができあがりました。いつものように描画用の各種データをセットし，rgnを(300, 200)に移動して表示すればdisplayは終了です。

＊

グラフマンのドキュメントをベースに，C言語でグラフマンを使う方法を模索してみました。ここではリスト4でテキスト4プレーンを使用する方法を示し，以後それを使ってプログラムしてきましたが，GMInitPaletを呼び出さなければ従来どおりの文字2プレーン，システム2プレーンの構成で扱うこともできます。このときはG_WHITEなどのマクロの代わりに，0～3のカラーコードを直接指定して描画するといいでしょう。

リスト5ではグラフマンがサポートしていないリージョンの直接生成まで行ったため，キャストやポインタが嵐のように登場してC初級・中級の方には難しかったかもしれません。勉強材料だと思って発奮してみてください。ここでは取り上げませんでしたが，グラフマンが現在のグラフィックカーソルの位置から線を引くという特徴を利用して，タートルグラフィックライブラリを構築することも可能です。

グラフマンはまだまだ多くの関数を抱えています。GMOpenRgnなどを使うと，さらに複雑なリージョンも作れるのかもしれません。楽しく遊べそうです。

リスト7　正統派のリージョン描画

```
 1: #include <stdio.h>
 2: #include <stdlib.h>
 3: #include <sxlib.h>
 4:
 5: #define HeapSize     0x20000
 6: #define X_Y(x,y)     (x)*0x10000+(y)
 7: #ifdef __GNUC__
 8:     asm( ".xdef _HEAP_SIZE" );
 9:     asm( "_HEAP_SIZE equ 204800");
10: #endif
11:
12: /*
13:  *    プロトタイプ宣言
14:  */
15:
16: int  SUPER( int );
17: int  C_CUROFF(void);
18: int  C_CURON(void);
19: int  C_LOCATE(int, int);
20: int  C_WIDTH(int);
21: int  C_FNKMOD(int);
22: void TXFILL( unsigned short * );
23:
24: void display( void );
25:
26: /*
27:  *    大域変数はリスト5と
28:  *    メインルーチンはリスト4と同じ
29:  */
30:
31: void display( void )
32: {
33:     /*
34:      *    リージョンの組み合わせで
35:      *    新しいリージョンを作る
36:      */
37:     region  **rgn, **rgn1, **rgn2;
38:
39:     GMOpenGraph( G_TXT, &t_graph );
40:     GMAPage( 15 );
41:
42:     GMBackColor( G_BLACK );
43:     GMPenMode( G_BACK + G_PSET );
44:     GMFillRect( &scr_rect );
45:
46:     /*
47:      *    作成するリージョン
48:      */
49:     rgn = GMNewRgn();
50:     (*rgn)->bounds.right = 100;
51:     (*rgn)->bounds.bottom = 100;
52:
53:     /*
54:      *    作成補助用のリージョン
55:      */
56:     rgn1 = GMNewRgn();
57:     (*rgn1)->bounds.right = 25;
58:     (*rgn1)->bounds.bottom = 25;
59:
60:     rgn2 = GMNewRgn();
61:     (*rgn2)->bounds.right = 50;
62:     (*rgn2)->bounds.bottom = 50;
63:
64:     /*
65:      *    XORは重なった部分を取り除く
66:      */
67:     GMXorRgn( rgn, rgn, rgn1 );
68:     GMMoveRgn( rgn2, X_Y( 25, 25 ));
69:     GMXorRgn( rgn, rgn, rgn2 );
70:     GMMoveRgn( rgn1, X_Y( 75, 75 ));
71:     GMXorRgn( rgn, rgn, rgn1 );
72:
73:     GMForeColor( G_RED );
74:     GMBackColor( G_YELLOW );
75:     GMPenMode( G_PPAT + G_PSET );
76:     GMPenPat( ppat );
77:     GMPenSize( X_Y( 4, 4 ));
78:
79:     GMMoveRgn( rgn, X_Y( 300, 200 ));
80:     GMFillRgn( rgn );
81: }
```

SXLIFE Part II

ポップアップメニューの追加

Nakamori Akira 中森 章

前回、ともかく何かを作ってみようということでお届けしたのがディスクにも入っていたライフゲームです。そこで今月から数回にわたって、SX-WINDOW用のソフトらしくするための機能を加えていきましょう。最初はポップアップメニューです。

100%のSXLIFE

先月のオマケディスクに入っていたSXLIFEはもう動かしてもらえたでしょうか。これは私がSX-WINDOW上のアプリケーションとして作った記念すべき第1号なのです。しかし，あの時点では時間的制約と私の不勉強のせいで最初イメージしていた100%のSXLIFEにはなりませんでした。そこで，SXFILEを真の姿に近づけていこうというのが本稿の意図です。

ところで，オマケディスク用のプログラムを編集部に渡した直後，UNIXのJUNETでX68000のニュースグループに奇しくも同じSX-WINDOW上のライフゲームが投稿されていたので，ライバル意識を持って実行してみたところ非常にショックを受けてしまいました。とにかく動きが高速なのです。確かに最高速で動作させると他のウインドウの動きが重くなってくるのですが，そんなのは言い訳になりません。このままでは悔しいので私のSXLIFEも今ではより高速に改造してしまいました。

今回の連載ではこの高速化技法については特に説明しませんが，要点をいうと1行処理をさらに高速化して処理単位を10行にしたということです。結果としてJUNET版と同程度の速さになったと思います（他のウィンドウ処理もそれほど重くなっていません)。興味のある人は改造してみてください。

さて，100%のSXLIFEとはなんでしょう。ここですべてを明かすのは面白くありませんから，今月，来月と小出しにしていくことにします（私自身もそれを実現できるかどうかはこれからの勉強次第なので大口はたたきたくない)。とりあえず，今月ではポップアップメニュー処理を追加します。

メニューを使う手順

オマケディスクのSXLIFEはドットの初期設定を右ボタンで行うようにしました。単純なドットの設定と消去のみをサポートしていたのみですが，本来なら右ボタンはポップアップメニュー（名前が長ったらしいので以下では単にメニューといいます）の項目選択に割り当てたいものです。そして，ドットの設定と消去のほかにもSXLIFEのバージョン説明や画面の全消去ぐらいの選択肢を持たせるべきだったと思っています。というわけで，さっそくこの機能を実現することにします。

例によって，いろいろなアプリケーションのソースや実行形式オブジェクトの逆アセンブルリストを必死に解析したところ，メニュー機能の実現には次の3つのステップを踏んでいることがわかりました。

- ●メニューの初期化
- ●メニューの表示と項目選択
- ●プログラム終了時の処理

これらの手順をSXLIFEに取り込んでやればよいのです。それぞれの項目について実際の手順を説明しましょう。

メニューの初期化

メニューの初期化とはメニューの実体を作成することです。そのためにはメニューを実現するために十分なだけ（20バイト＋α）のメモリ領域を確保して，そこにメニュー情報を書き込めばおしまいです。このメニュー情報として次のようなものを決定しなければなりません。

●メニューID（オフセット0）

0か1を設定します。これはメニュー定義関数（あとで説明）のリソースIDです。

●メニュー定義関数へのハンドル（オフセット6）

ハンドルとはC言語でいうポインタのポインタみたいなもので，その指し示す先にあるデータが実体へのアドレスとなっています。メニュー定義関数とは表示するメニューの形状を決定するリソースで，

　　__RMRscGet

というSXコールで獲得できます。そのときには，第1引数（あとからスタックに積まれるほう）を'MDEF'という32ビット文字定数に，第2引数（先にスタックに積まれるほう）を上で出てきた16ビットのメニューID（種類）にします。メニューIDとして現在は0と1のみしかサポートしてないようです。0がタイトルなしメニュー（図1)，1がタイトルありメニュー(図2）のようです。

●メニューアイテムの許可フラグ（オフセット10)

メニューアイテムとはメニューの中に表示されている文字列のことです。マウスカーソルを持って行くとそこが反転してメニューが選択されていることを示すアレのことです（図3)。SX-WINDOWでは（Macintoshと同じく)，たとえメニューの中にあるアイテムでも，状況によってはそれを選択することができないように設定することができるようになっています（このとき選択できないアイテムは他のアイテムより薄く表示される)。それを実現するのがこの許可フラグです。

メニューアイテムにはメニュー内で上から順に1，2，3，……と番号が付けられていて，その番号に対応するビット位置を

0にした32ビットデータが許可フラグになります。したがってすべてのアイテムが選択可能なメニューでは許可フラグはオール1になります。また，許可フラグのビット0には意味がないようです。また，許可フラグのビット数から推測できるように，メニューアイテムの最大は31個になります。

●メニュー定義関数のデータ（オフセット14）

タイトルありメニューを使用するとき，メニューのタイトルが格納されているメモリ上のアドレスを設定します。メニューのタイトルは最初の1バイトがタイトルの長さ（バイト数）を表し，2バイト目以降が文字列です。もちろん，タイトルなしメニューの場合は設定する必要はありません。

●メニューアイテム数（オフセット18）

メニューアイテムの総数から1を引いた値を設定します。

＊

メニューを使用する場合，少なくとも以上のメニュー情報を設定しなければなりません（各情報の正確なバイト数はドキュメントを見てください）。これが先に20バイト＋αと述べたうちの20バイト分で，残りの＋αはこれに続くメニューのアイテムデータ情報です。1つのアイテムデータは次のような構造をしています。

●ショートカットコード（1バイト）

ASCIIコードの1文字です。これはメニューアイテムを選択したのと同じ効果をキーボードのキー入力で実現するためにはどのキーを押したらいいのかを表示するための文字です。ここに41_H（'A'）と書かれていたらAキーを押したときにそのメニューアイテムを選択したのと同じことになります。ただし，プログラムをそのように（キー入力対応に）書いておかなければまったく無意味です。ショートカットコードが必要ないときはここに0を書いておきます。

●チェックマーク（1バイト）

ここに0以外を書いておくとメニューアイテムを表示するとき，横にそのアイテムがすでに選択されていることを示すチェックマーク（'レ'というような記号）が同時に表示されます。これは単なる表示にすぎませんからチェックが付いたメニューアイテムがどのような意味を持つかはプログラム任せです。

●表示文字列

メニューアイテムとして表示する文字列です。最初の1バイトが文字列のバイト数を示し，それ以降が実際の文字列です。

＊

このメニューアイテム情報がメニューアイテムの個数だけ続くのです。ここで注意すべきことは，ショートカットコード，チェックマーク，表示文字列の長さの合計が偶数バイトになるようにしなければならないという点です。つまり，全体が偶数バイトになるように表示文字列の長さを調整しなくてはなりません。これに違反するとメニューを表示するときにアドレスエラーを起こしたり，グチャグチャなメニューが表示されたりします。

最後になりましたが，メニューのためのメモリ領域を確保するためにもSXコールを使用します。それは，

__MMChHdlNew

というSXコールで，引数としてメニューに必要な20バイト＋αのバイト数を与えます。最初のうち，私はメモリ領域をSXコールでわざわざ確保しなくてもプログラム内のデータ領域を使用すればうまくいく（もちろんメモリ領域へのハンドルもどきを作る）と考えていました。しかし，それをやると全然メニューが表示されなかったので，あっさりと他のプログラムで行っているメモリを確保する方式に乗り換えてしまったことを白状しておきましょう（そうするとうまくいった）。

以上がメニューの初期化プログラムのすべてです。私が実際に書いた初期化プログラムをリスト1に示しておきましょう。ほとんどコメントを付けていませんが何をしているかはわかりますね。ここではメニューの初期化時にエラーが発生してもなにくわぬ顔をして処理を終了しています。これは，たとえメニューが使えなくてもライフゲームを実行させようというためです（単なる手抜きという意味もあったりして）。ただ，エラーがあったかどうかは__useMenu（a5）という変数の値でわかるようになっています（プログラム終了時に参照する）。

なお，リスト1のinitMenu関数はウィンドウをオープンしたあと，ライフゲームの初期化を行う直前で呼べばいいでしょう。

メニューの表示と項目選択

いよいよメニュー処理のハイライト（ウィンドウのハイライト表示とは無関係ですよ）であるメニューの表示と項目選択です。これは，メニューを表示したあと，マウスカーソルを追い続け，マウスカーソルがあるメニューアイテムを指していたらそこの表示色を反転し，メニューアイテムを指しながらマウスの（右）ボタンが離されたら

図1　タイトルなしメニュー

ラーメン
ギョーザ
レバニライタメ
チャーシュー
メンマ
シューマイ

図2　タイトル付きメニュー

正義の味方
ウルトラマン
仮面ライダー
ポワトリン
キカイダー
デビルマン
イコちゃん

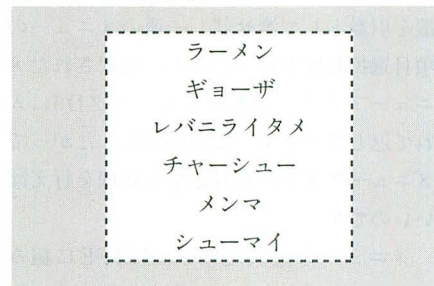
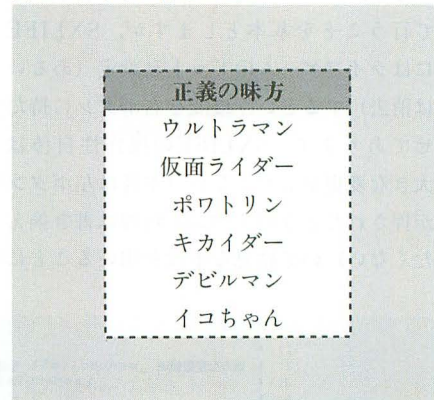
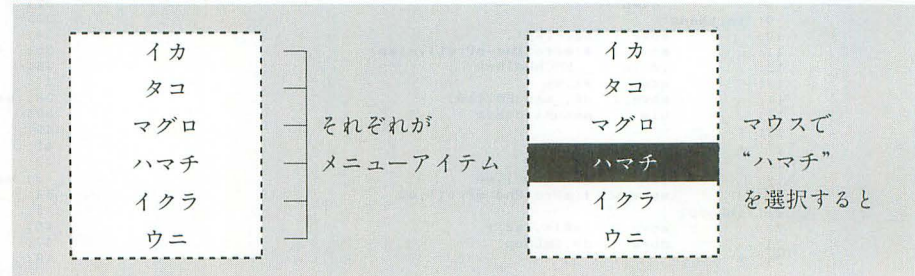
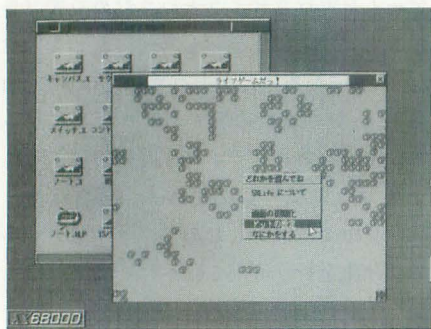



図3　メニューアイテム

ポップアップメニューの例

そのアイテム番号をプログラムに知らせる，という複雑な処理をしなくてはなりません。考えただけで絶望的な気持ちになってきますね。でも心配はいりません。これら一連の処理を行ってくれるSXコールがSX-WINDOWでは用意されています。それが，

__MNSelect

というSXコールです。これにメニューへのハンドル（__MMChHdlNewで求めた値）と，マウスの右ボタンが押された位置を引数として渡せば，一連のメニューの項目選択処理を行ったあと，選択されたメニューアイテムの番号をレジスタD0に入れて返してきます。この番号にしたがってメニューアイテムに対応する処理を行えばいいのです。

メニューの選択処理をSXLIFEに組み込むにはちょっとした工夫が必要です。なぜなら，メニュー処理はマウスの右ボタンで行うことを基本としますが，SXLIFEにはライフゲームのドットを設定（あるいは消去）するという機能を右ボタンに持たせてあります。SXLIFEの操作性自体は大きな変更をしたくない（本音は左ボタンが押されたときのイベント処理は書き換えたくない）ので姑息な手段を用いることにします。つまり，ドットを設定するモードにあるとき，すなわちuserWRK(a5)という変数の値が0でないときは今までどおりドットの設定・消去処理を行います。

userWRK(a5)の値が0であるとき（これが初期値）は__MNSelectでメニュー処理を行います。古い版のSXLIFEではマウスの右ボタンが押されると無条件にuserWRK(a5)の値は1になりましたが，新版のSXLIFEではメニューによって「ドットの設定」が選択されたときにuserWRK(a5)が1になります。userWRK(a5)を0に戻すのは古い版と同じくマウスの左ボタンが押されたときです。こうすることで古い版に対して最小の修正量でSXLIFEにメニュー処理を組み込むことができます。

リスト2にマウスの右ボタンが押されたときのイベント処理のすべてを示します。古い版のSXLIFEに対してどこが変更になったかはわかりますね。

リスト2で行っている選択されたメニューアイテム（__MNSelectの返り値）に対する処理をまとめると次のようになります（リスト1のメニューアイテム定義も併せて見てくださいね）。

0…項目選択なし（何もしない）
1…SXLIFEについての情報を表示
2…許可フラグで禁止されているので選択されない（単なる飾り）
3…画面を初期化（ドットの全消去）
4…ドットの設定・消去（従来どおり）
5…何かをする（実は何もしない）

といったぐあいです。メニューアイテム3の画面の初期化処理（initMap）が少しややこしい（結構技巧的な処理をやっている）けれど全体の流れはわかるでしょう。

プログラム終了時の処理

プログラムを終了するときには，SXコールで確保したメニューのための領域を解放（捨てる）することが必要になります。このためには，

__MMHdlDispose

というSXコールを使います（リスト1でもメニュー定義関数が得られなかった場合ひっそりとこのSXコールを行っていたことに気づきましたか）。このSXコールの引数にはメニューへのハンドル（__MMChHdlNewで求めた値）を与えます。ただ，メニューの初期化が失敗しているときは，このような処理は必要ありません（やるとかえって悪い）。具体的には変数_useMenu(a5)の値が0のときはSXコールを発行せず，1のときのみ発行するようにしています。このリストはあまりにも簡単ですから必要ないでしょう。

さて来月は

メニュー処理の追加は思っていたよりも簡単にできてしまったというのが実感です。まあ，メニュー処理は100%のSXLIFEのための過程でしかありませんから当然といえば当然でしょう。私が真にやりたかったのはもっと別のことなのです。それを実現するためにはメニューによる行動選択が是非とも必要になってくるので，今月はワンクッション置いてみました。さて，来月はベールを脱ぐと思われる100%のSXLIFEとはなんなのでしょう。乞うご期待というところですね。

リスト1

```
 1: *
 2: * 新たな変数領域 _menuHdl(a5) を追加
 3: *                _useMenu(a5)
 4: *
 5: *********************
 6: * メニューの初期化
 7: *********************
 8:         .even
 9: initMenu:
10:         move.l  a2,-(sp)
11:         move.l  #(mProtEnd-mProt),-(sp)
12:         .dc.w   __MMChHdlNew
13:         addq.l  #4,sp
14:         move.l  d0,_menuHdl(a5)
15:         ble     menuAllocErr
16: *
17:         move.l  d0,a2
18:         move.l  (a2),a2
19:         lea     mProt(pc),a0
20:         move.w  #(mProtEnd-mProt),d0
21: imLoop:
22:         move.b  (a0)+,(a2)+
23:         dbra    d0,imLoop
24: *
25:         clr.w   -(sp)
26:         move.l  #'MDEF',-(sp)
27:         .dc.w   __RMRscGet
28:         addq.l  #6,sp
29:         tst.l   d0
30:         bmi     menuRsGetErr
31:         move.l  _menuHdl(a5),a2
32:         move.l  (a2),a2
33:         move.l  a0,6(a2)
34:         move.b  #1,_useMenu(a5)
35:         move.l  (sp)+,a2
36:         rts
37:
38: menuAllocErr:
39:         move.b  #0,_useMenu(a5)
40:         move.l  (sp)+,a2
41:         rts
42:
43: menuRsGetErr:
44:         move.l  _menuHdl(a5),-(sp)
45:         .dc.w   __MMHdlDispose
46:         addq.l  #4,sp
47:         move.b  #0,_useMenu(a5)
48:         move.l  (sp)+,a2
```

```
49:         rts
50:
51: *
52: * メニュー原型（20バイト）
53: *
54:         .even
55: mProt:
56:         .dc.w   0
57:         .dc.w   0
58:         .dc.w   0
59:         .dc.l   0
60:         .dc.l   $fffffffb       ;; 許可フラグ
61:         .dc.l   0
```

```
62:         .dc.w   4                       ;; アイテム数－1
63: *
64: * メニューアイテム
65: *
66: mItem:
67:         .dc.b   $00,$00,$0f,'SXLife について'   ;; 18 Byte
68:         .dc.b   $00,$00,$09,'.........'         ;; 12 Byte
69:         .dc.b   $00,$00,$0d,'画面の初期化 '     ;; 16 Byte
70:         .dc.b   $00,$00,$0d,'ドットの設定 '     ;; 16 Byte
71:         .dc.b   $00,$00,$0d,'なにかをする '     ;; 16 Byte
72:         .dc.b   $00
73: mProtEnd:
```

リスト2

```
  1: ************************************************
  2: * マウスダウン・イベント時の処理（右ボタン）
  3: ************************************************
  4:         .even
  5: EV_MSRDOWN:
  6:         movem.l d1-d7/a1-a5,-(sp)       ;; レジスタを保存
  7: *
  8: * 自分のウインドウかを調べる
  9: *
 10:         move.l  evntRec+with(a5),d0
 11:         beq     retMSRDOWN              ;; どのウインドウにも該当しない
 12:         cmp.l   wPointer(a5),d0
 13:         bne     retMSRDOWN              ;; 他のウインドウだった
 14: *
 15: * カレントポートにセット
 16: *
 17:         move.l  d0,-(sp)
 18:         .dc.w   __GMSetGraph
 19:         addq.l  #4,sp
 20:         move.l  a0,a2
 21:         tst.w   active(a5)
 22:         bne     procMSRDOWN             ;; 前からアクティブだった
 23:         move.w  #1,active(a5)
 24:         move.w  #$ffff,evntMsk(a5)      ;; イベントマスクをセット
 25: *
 26: * ウインドウを切り替える
 27: *
 28:         pea     (a2)
 29:         .dc.w   __WMSelect              ;; 自分をセレクト
 30:         addq.l  #4,sp
 31:         .dc.w   __EMRStill              ;; 右ボタンが押されたままか
 32:         tst.l   d0
 33:         beq     noStillMSRDOWN
 34:         move.l  evntRec+where(a5),-(sp)
 35:         .dc.w   __WMFind                ;; 座標がウインドウのどこにあるか
 36:         addq.l  #4,sp
 37:         cmp.w   #inContent,d0
 38:         bne     noStillMSRDOWN          ;; コンテントリージョン以外
 39: *
 40: * マウスの座標を知る
 41: *
 42: procMSRDOWN:
 43:         tst.b   userWRK(a5)             ;; 設定モードか
 44:         bne     dotSetReset
 45: **
 46: ** メニューを表示する
 47: **
 48:         move.l  evntRec+where(a5),-(sp) ;; ここからがメニュー処理
 49:         move.l  _menuHdl(a5),-(sp)
 50:         .dc.w   __MNSelect
 51:         addq.l  #8,sp
 52:         tst.b   d0
 53:         beq     doMenu0
 54:         cmpi.b  #3,d0
 55:         blt     doMenu1
 56:         beq     doMenu3
 57:         cmpi.b  #5,d0
 58:         blt     doMenu4
 59:         beq     doMenu5
 60:         bra     noAction
 61:
 62: doMenu0:
 63:         bra     noAction                ;; 選択されなかった
 64:
 65: doMenu1:
 66:         pea     aboutMess(pc)
 67:         move.w  #1,-(sp)
 68:         dc.w    __DMError
 69:         addq.l  #6,sp
 70:         bra     noAction
 71:
 72: aboutMess:
 73:         .dc.b   'ライフゲーム by 中森 章,',$0d
 74:         .dc.b   ' Nov. 28, 1990',0
 75:
 76:         .even
 77: doMenu3:
 78:         bsr     initMap
 79:         bra     noAction
 80:
 81: doMenu4:
 82:         move.b  #1,userWRK(a5)          ;; 描画をロックする
 83:         bra     noAction                ;; ロックの解除は左ボタン
 84:
 85: doMenu5:
 86:         bra     noAction
 87:
 88: dotSetReset:
 89:         .dc.w   __EMMSLoc
 90:         tst.l   d0
 91:         bmi     errMSRDOWN
 92:         move.l  d0,d1
 93:         moveq.l #0,d2
 94:         move.w  #0,d1
 95:         swap    d1                      ;; 水平
 96:         move.w  d0,d2                   ;; 垂直
 97:         divu    #12,d2
 98:         divu    #12,d1
 99:         move.w  d1,-(sp)
100:         move.w  d2,-(sp)
101:         lea     _field(a5),a1
102:         lsl.l   #6,d2
103:         lea     (a1,d2.w),a1
104:         lea     (a1,d1.w),a1
105:         move.l  wPointer(a5),-(sp)
106:         .dc.w   __GMSetGraph
107:         addq.l  #4,sp
108:         tst.b   (a1)
109:         beq     nonExist
110:         bsr     preset
111:         clr.b   (a1)
112:         bra     cmnMSRDWN
113: nonExist:
114:         bsr     pset
115:         move.b  #1,(a1)
116: cmnMSRDWN:
117:         addq.l  #4,sp
118: noAction:
119:         move.l  wPointer(a5),-(sp)      ;; グローボックスを書き直す
120:         .dc.w   __WMDrawGBox
121:         addq.l  #4,sp
122: *
123: * イベントレコードをのぞく
124: *
125: noStillMSRDOWN:
126:         pea     evntRec(a5)             ;; イベントレコード
127:         move.w  evntMsk(a5),-(sp)       ;; イベントマスク
128:         .dc.w   __TSGetEvent
129:         addq.l  #6,sp
130: *
131: * 今回の状態をセーブ
132: *
133: retMSRDOWN:
134:         movem.l (sp)+,d1-d7/a1-a5
135:         rts
136:
137: errMSRDOWN:
138:         moveq.l #-1,d0
139:         bra     retMSRDOWN
140:
141:         .even
142: initMap:
143:         movem.l d1/d3/d4/a1,-(sp)
144:         move.l  wPointer(a5),-(sp)
145:         .dc.w   __GMSetGraph
146:         addq.l  #4,sp
147: *
148:         move.w  #(64*DispHight-1),d1    ;; ループ回数
149:         moveq.l #0,d3                   ;; X座標
150:         moveq.l #0,d4                   ;; Y座標
151:         lea     _field(a5),a1
152: clrLoop:
153:         bclr.b  #0,(a1)+
154:         beq     clrSkip1
155:         move.w  d4,-(sp)
156:         move.w  d3,-(sp)
157:         bsr     preset2
158:         addq.l  #4,sp
159: clrSkip1:
160:         addi.w  #12,d4
161:         cmpi.w  #(64*12),d4
162:         bcs     clrSkip2
163:         addi.w  #12,d3
164:         move.w  #0,d4
165: clrSkip2:
166:         dbra    d1,clrLoop
167: *
168:         move.l  #1,_xhold(a5)           ;; 変数類の初期化
169:         move.l  #-1,_nv0(a5)
170:         lea     _vs0(a5),a1
171:         move.l  a1,_vp0(a5)
172:         lea     _vs1(a5),a1
173:         move.l  a1,_vp1(a5)
174:         lea     _ys0(a5),a1
175:         move.l  a1,_yp0(a5)
176:         lea     _ys1(a5),a1
177:         move.l  a1,_yp1(a5)
178: *
179:         move.l  wPointer(a5),-(sp)
180:         .dc.w   __WMDrawGBox            ;; グローボックス描き直し
181:         addq.l  #4,sp
182:         movem.l (sp)+,d1/d3/d4/a1
183:         rts
```

「SXエンターテイメントキット」計画

Ogikubo Kei 荻窪 圭

寒い日はガスファンヒーターをがんがんにつけて，バイクは転がさずにマウスを転がす。それが正しい日本の冬というもの。それにつけても，マウスのおともに欲しいものがありませんか？　そこで，ぼくらのSX-WINDOWに味わい深いスモールアプリケーションを用意したいナ，というお話です。

●とりあえず前置きを

最近，お家の事情があっていろんなパソコンのウィンドウシステムを触っているのだが，皆さんそれぞれ癖があって面白い。

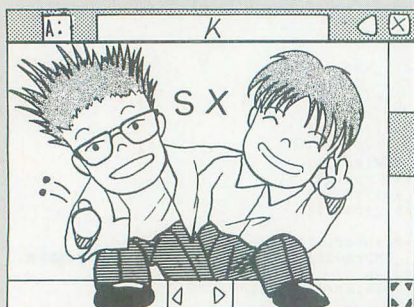
なかでもMacintoshがいちばん可も不可もなくまとまっている気がする。長年の蓄積があるからね。Macintoshが出てからもう6年もたったのだから，洗練されてなければ嘘になる。ただあれは，ウィンドウシステムのイメージとはちょっと違う。アプリケーション1つひとつが全画面を専有してしまうからだ。ウィンドウシステムというと，1アプリケーション1ウィンドウというイメージがある。ないか？　まあ，イメージの問題だ。

MS-WINDOWS 3.0やNeXTはそうなっている。アプリケーションは自分のウィンドウだけしかいじらないよ，ってことだ。NeXTになると，アプリケーション間通信ってな高度な技があっていいのだが，あいつは68040であって，68000のX68000より40も大きい。

もっとも，NeXTはUNIXマシンであって，パソコンというのもなんだが，ハイエンドPCとして売っていきたいようなことをキヤノンの人はいっていた。当のジョブズさんも，そうとらえているような気はする。

MS-WINDOWS3.0はまだ日本語版が出ていないのでよくわからないが，世間で騒いでいるほど使いやすいものではない。アプリケーションを動かすのは楽でいいが，インストールがやけに面倒だったり，プログラムマネージャとファイルマネージャが別々に存在していて，ファイル管理がうっとうしいなどの欠点を持っている。

SX-WINDOWはウィンドウのアイコン化ができない点を除けば，とてもいい線をいっていると思う。

まあ，この詳細な比較は来月号にでもレポートするとして，SX-WINDOWがどのウィンドウシステムにも勝てないところといえば，アプリケーションがないことだ。

SX-WINDOWでもソフトが欲しい。そういうわけである。

●ウィンドウアプリの原点は？

まず何から始めようか。「大人のためのX68000」などというあやしい連載を抱える身としては，一応，ビジネスソフト（と一般的にいわれている類のソフト）が欲しいといっておくべきだろうが，ゲーマーとしての私はやはりアミューズメント系のウィンドウが欲しいのである。こういうときは「大人のためのSX-WINDOW用ゲーム」が欲しいといっておけば事足りるということになっているので，そういっておこう。しぶいウィンドウデザインには大人のためのゲームが似合うのだ。へへへ。アダルトゲームといってもいいぞ。えへへへ。

さて，「SXエンターテイメントキット」というのは，SX-WINDOW用の小さなゲームがいっぱい詰まったディスクのことである。米マイクロソフト社がMS-WINDOWS用に出している「MSエンターテイメントキット」から名前を拝借した。このディスクには，小さなウィンドウひとつで済むようなちょっとしたゲームがいくつも収められている。「Hex TETRIS」なんていう，四角ではなく六角形のブロックを使うやつも収められているが，傑作はマインスイーパーっていうパズルだ。

暇潰しに最適な，ちょっとしたソフト。それでもって，そいつに収められたソフトはみな他愛ないもの。が，Yet Another Columnにみんながハマったように，ああいう単純なゲームにこそ真髄が隠されているのだ。今こそ，原点に帰るべきだ。

●要するにゲーム集

「SXエンターテイメントキット計画」は，そういうソフトをOh!Xで用意しようという温かい計画である。あくまでも計画である。用意できるかどうかはまったくわからない。

なぜなら，キットに収めるゲームは，読者から募集するからである。私は作らないで(作ってみたいとは思っているが)，ただ，投稿を待っているのである。

どんなソフトが向いているか。

小さなウィンドウで済むもの。立ち上げたまま，いつもは裏に隠しておき，やりたくなったらアクティブにする。

コラムズなんて最適。

ぼおっと，マウスひとつで遊べるもの。

先月ディスクに入っていた，クロンダイクなんかはそうだ。そのテのカードゲームもいいかもしれない。ポーカーなんかのギャンブル系ソフトも見逃せない。競馬ソフトを作って，実際のレースは裏でマルチタスク，なんてのも楽しそうだ。競馬はつねに行われており，ときどき，競馬ウィンドウをアクティブにして，次のレースの馬券を買うのだ。

反応速度に結果が左右されないもの。

マスターマインド。じっくり頭を使うやつだね。リバーシでもいい。

ショートプロなもの。伝統的な用語を使うなら，ピコピコゲームなもの。昔懐かしいゲームの復活の場とするのもよかろう。

目いっぱいキッチュに，ウィンドウを52個開いて，神経衰弱をするとか，百枚開いて，百人一首するとか，ばかばかしいのもいいかもしれない。

さらにその上をいって，ポコポコ開くウィンドウを探してそこをクリックするモグラたたきとか（失敗してほかのウィンドウを開いてしまったら，そいつがアクティブになって，肝心のモグラウィンドウが見えなくなって困るっていうのもいい）。

個人的には，早撃ちゲームなんかもいいな。ウィンドウの中に扉がいくつかあって，どこかが不意に開く。出てきたのが悪人だったら，マウスカーソル（もちろん照準の形だ）を動かして撃つ。遅いと撃たれる。このテのやつが結構好きだったりするのだ。

＊

とりあえず，些細なものでもできたら投稿する。これが始まりである。

始まりがあるからといって終わりがあるわけでもないのだが，いいものが集まったら，なんらかの形でディスクに収め，「SXエンターテイメントキット」として世に出すことになろう。

その中身は多くのSX-WINDOWユーザーのハードディスクに収められ，わずかな暇の相手となるだろう。

受験生も社会人もサラリーマンもがーんば。

X68000CARD DRV用カードゲーム

スロットポーカー

Hanyu Junya 羽生 純也

お待たせしました。いよいよトランプゲームの真打ち，ポーカーの登場です。ボーナスゲームで得点倍増にチャレンジすれば一獲千金も夢じゃないぞ。今回はひとり遊びですが，コンピュータとの対戦版もほしいですね。誰か挑戦する人はいませんか？

ひとり遊び用のポーカーです。

X-BASICからリスト1を入力してください。実行の際にはCARD.FNCまたはCARD2.FNCの拡張されている必要があります。CARD.FNCをお持ちの方はもうお馴染みでしょうから，ここでは1月号で発表されたCARD2.FNCの組み込みについて解説しておきましょう。

まず，1991年1月号の付録ディスクを展開したものを用意します。ディスク3の中にCARD2というディレクトリがあります。コマンドシェルを立ち上げ，

CD CARD2

を行ってください。

CARDDRV TR.DAT

でドライバを組み込みます。

次にBASICのあるディクトリに移動して，

COPY B：CARD2.FNC

のようにしてCARD2.FNCをコピーします。あとはBASIC.CNFにCARD2.FNCを加えて，CARD2.FNCを組み込んでください。これでCARDDRVが使用できます。

ポーカーとは

さて，あまりにも有名なゲームですからポーカーのルールについては特に解説する必要はないでしょう。

トランプゲームのなかでは比較的歴史が浅いにもかかわらず，ポーカーはカードギャンブルの代表格として扱われています。

「ポーカーという名前が初めて記録されたのは1829年である。ポーカーはニューオリンズに発生して，スチームボートの酒場とともにミシシッピ河をさかのぼっていった（トランプゲーム大百科より抜粋）」

ポーカーの起源としてペルシアのカードゲーム「アスナス(As Nas)」や長い歴史を持つ「ブラッグ(Brag)」が挙げられることもありますが，定かではありません。アメリカで発生したというのは確かなようです。

初期のストレートポーカー，配られた5枚のカードで役を競うものから，手札のいくつかを交換（ドロー）できるドローポーカーが完成されました。ふつうにポーカーといえばドローポーカーを指しますが，さらにギャンブル志向のスタッドポーカー，カード交換はなしで伏せられた1枚のカードに1枚ずつオープンカードを加えていくタイプのポーカーなども発生しています。あまりにも多様に変化するのでいまだに公式ルールを持っていません。いずれにせよ，「ポーカーフェイス」というように，技術よりもハッタリがこのゲームの信条でしょう。

ちなみに「ポーカーという名前はフランスのゲームポーク（Poque）からくるとも，または「イッヒ　ポッヘ(Ich Poche)」といってプレイヤーがテーブルを叩くことによってパスを表すドイツのゲーム，ポッホシュピール（Pochspiel）に由来するとも考えられる」そうです。

今回のゲームについて

今回のプログラムはひとり遊び用ですから，もっぱら役作りを目的としたゲームになります。要領はゲームセンターにあるスロットポーカーと同様です。配られたカードのうちHOLDするものを指定してカードを交換し，できた役の強さで賭け金の倍率が変わります。

役はふつうのポーカーハンドからワンペアを除いたものが有効で，役が成立するとハイ＆ローのボーナスゲームにチャレンジできます。成功すれば配当が2倍，失敗すれば配当はなし。持ち金が10000を超えるとゲームクリアとなります。

なお，一部の役の判定が通常のポーカーとは異なります。フラッシュとストレートの部分ですね。ポーカーではふつう，ストレートフラッシュ，フォーカード，フルハウス，フラッシュ，ストレート，スリーカード，ツーペア，ワンペア，ブタという順の役の強さになっています。これらが配牌時（？）に成立する確立はそれぞれ，0.0015％，0.024％，0.144％，0.1956％，0.392％，2.15％，4.75％，42.2％，50.1％となっています。これでいけば，フラッシュはストレートより強い役であっても不思議はないのですが，配牌で大きな役が成立することを期待してもしかたありません。その役を狙う気になるところ（麻雀でいえば聴牌時）からの成功確率ではフラッシュとストレートは逆転しますから，このほうが実用的といえるかもしれません。

この作品が投稿されてきたときにはCARD2.FNCもCARDDRV.Xもありませんでしたが，今回試してみるとそのままでコンパイルできました。これまで発表されたカードゲームの多くが一部変更しないとコンパイルできないことを思えば，非常に素直にプログラムされている作品だといえます。

ただし，コンパイルするときは適当な個所に適当にウエイトを入れてください（そのままでも遊べますが）。

リスト1

```
  10 /*
  20 /* THE_POKER
  30 /*
  40 int kin,kake,up,max
  50 int c_cnt,round
  60 int mx,my,x,y,lb,rb
  70 str pasa="@59v15c",atari="@56v15o4120aeae",wake="@56v15
c"
  80 dim str yaku(7)={"ROYAL STRAIGHT FLUSH",
  90                  "STRAIGHT FLUSH",
 100                  "FOUR CARD",
 110                  "FULL HOUSE",
 120                  "STRAIGHT",
 130                  "FLUSH",
 140                  "THREE CARD",
 150                  "TWO PAIR"}
 160 dim int ritsu(7)={500,100,40,10,7,5,3,2}
 170 dim int zahyou(4)={88,160,232,304,376}
 180 dim int tmp(5,1),cc(5),c(51)
 190 dim str m(12)
 200 init()
 210 nv_init()
 220 gamen()
 230 jyunbi()
 240 while 1
 250   repeat
 260     shuffle()
 270     c_hyouji(0)
 280     bet()
 290     game()
 300     round=round+1
 310   until kin=0 or kin>=10000
 320   over()
 330 endwhile
 340 end
 350 /*
 360 func bet()                  /* お金を賭ける
 370   int i
 380   s_hyouji(1,7,8)
 390   if kin>=1000 then up=100 else up=10
 400   if kin<=100 then max=50 else max=kin/2
 410   repeat
 420     msstat(x,y,lb,rb)
 430     if lb=-1 and rb=0 and kake+up<=max and kin-up>=0 th
en {
 440       kin=kin-up:m_hyouji()
 450       kake=kake+up:k_hyouji()
 460       if kake<=100 or (kin>=1000 and kake<=1000) then w
ait(2)
 470     }
 480     if lb=-1 and rb=-1 then {
 490       kin=kin+kake:m_hyouji()
 500       kake=0:k_hyouji()
 510     }
 520   until lb=0 and rb=-1 and kake>0
 530   for i=0 to 7
 540     y_hyouji(i,0)
 550   next
 560   s_hyouji(0,0,0)
 570 endfunc
 580 /*
 590 func game()                 /* ゲームをする
 600   int i,rank
 610   c_hyouji(1)
 620   s_hyouji(2,9,8)
 630   rank=hantei()
 640   if rank<>8 then y_hyouji(rank,1)
 650   repeat
 660     mspos(mx,my)
 670     msstat(x,y,lb,rb)
 680     if lb=-1 then {
 690       for i=0 to 4
 700         if mx>=zahyou(i) and mx<=zahyou(i)+47 then {
 710           hold(i)
 720           break
 730         }
 740       next
 750       msbtn(0,0,0)
 760     }
 770   until rb=-1
 780   for i=0 to 4
 790     if cc(i)=0 then {
 800       c_put(88+72*i,256,0):wait(2)
 810     }
 820   next
 830   for i=0 to 4
 840     if cc(i)=0 then {
 850       c_put(88+i*72,256,c(c_cnt)):oto(pasa)
 860       cc(5)=c(i):c(i)=c(c_cnt):c(c_cnt)=cc(5)
 870       c_cnt=c_cnt+1:wait(2)
 880     }
 890   next
 900   if rank<>8 then y_hyouji(rank,0)
 910   rank=hantei()
 920   if rank<>8 then {
 930     y hyouji(rank,1)
 940     kake=ritsu(rank)*kake:k_hyouji()
 950     double()
 960   }
 970   kake=0:k_hyouji()
 980   s_hyouji(0,0,0)
 990   locate 12,23:print space$(40)
1000   jyunbi()
1010 endfunc
1020 /*
1030 func hold(no;int)           /* 取っておくか調べる
1040   if cc(no)=0 then {
1050     cc(no)=1
1060     locate 12+no*9,23:print "HOLD"
1070   } else {
1080     cc(no)=0
1090     locate 12+no*9,23:print "    "
1100   }
1110 endfunc
1120 /*
1130 func hantei()               /* 役の判定
1140   int i,j
1150   for i=0 to 4
1160     tmp(i,0)=c(i)/13
1170     tmp(i,1)=c(i) mod 13
1180     if tmp(i,1)=0 then tmp(i,0)=tmp(i,0)-1:tmp(i,1)=13
1190   next
1200   for i=0 to 3
1210     for j=i+1 to 4
1220       if tmp(i,1)>tmp(j,1) then {
1230         tmp(5,0)=tmp(i,0):tmp(5,1)=tmp(i,1)
1240         tmp(i,0)=tmp(j,0):tmp(i,1)=tmp(j,1)
1250         tmp(j,0)=tmp(5,0):tmp(j,1)=tmp(5,1)
1260       }
1270     next
1280   next
1290   if mark()=1 then {
1300     if tmp(0,1)=1 and tmp(1,1)=10 and tmp(2,1)=11 and t
mp(3,1)=12 and tmp(4,1)=13 then {
1310       return(0)
1320     }
1330     if strate()=1 then return(1)
1340     return(5)
1350   }
1360   if strate()=1 then return(4)
1370   switch same(0)
1380     case 3:return(2)
1390     case 2:if same(3)=1 then return(3)
1400            return(6)
1410     case 1:switch same(2)
1420              case 2:return(3)
1430              case 1:return(7)
1440              case 0:if same(3)=1 then return(7)
1450                     return(8)
1460            endswitch
1470     case 0:switch same(1)
1480              case 3:return(2)
1490              case 2:return(6)
1500              case 1:if same(3)=1 then return(7)
1510                     return(8)
1520              case 0:if same(2)=2 then return(6)
1530                     return(8)
1540            endswitch
1550   endswitch
1560 endfunc
1570 /*
1580 func mark()                 /* マークが揃っているか調べる
1590   int i
1600   for i=0 to 3
1610     if tmp(i,0)<>tmp(i+1,0) then return(0)
1620   next
1630   return(1)
1640 endfunc
1650 /*
1660 func strate()               /* 数字が並んでいるか調べる
1670   int i
1680   for i=0 to 3
1690     if abs(tmp(i,1)-tmp(i+1,1))<>1 then return(0)
1700   next
1710   if abs(tmp(0,1)-tmp(4,1))=4 then return(1)
1720   return(0)
1730 endfunc
1740 /*
1750 func same(st;int)           /* 同じ数字が何枚あるか調べる
1760   int i,cnt=0
1770   for i=st+1 to 4
1780     if tmp(st,1)=tmp(i,1) then cnt=cnt+1
1790   next
1800   return(cnt)
1810 endfunc
1820 /*
1830 func double()               /* ボーナスゲームをするか
1840   int end_f=0
1850   s_hyouji(0,0,0)
1860   s_hyouji(3,4,10)
1870   repeat
```


▶NETを開局してわかったんですけど，NETの運営管理って大変なんですね。でも，ちらほらと会員が増えてボードにいっぱい書き込みがあると，とてもうれしくなってしまう今日この頃です。　小板橋　敏弘(21)福島県

```
1880     msstat(x,y,lb,rb)
1890     if lb=-1 then end_f=1
1900     while end_f=1
1910       end_f=bonus()
1920     endwhile
1930   until rb=-1
1940   kin=kin+kake:m_hyouji()
1950 endfunc
1960 /*
1970 func bonus()                   /* ボーナスゲーム
1980   int c1,c2
1990   s_hyouji(0,0,0)
2000   locate 12,23:print space$(40)
2010   fill(88,256,423,351,8)
2020   c_put(160,256,c(c_cnt)):oto(pasa):wait(2):c1=check(c(
c_cnt)):c_cnt=c_cnt+1
2030   c_put(304,256,0):oto(pasa):wait(2)
2040   s_hyouji(3,11,12)
2050   wait(5)
2060   click()
2070   c_put(304,256,c(c_cnt)):oto(pasa):wait(2):c2=check(c(
c_cnt))
2080   if (lb=-1 and c1<c2) or (rb=-1 and c1>c2) then {
2090     oto(atari)
2100     kake=kake*2:k_hyouji()
2110     s_hyouji(0,0,0)
2120     s_hyouji(6,5,10)
2130     click()
2140     if lb=-1 then c_cnt=c_cnt+1:return(1)
2150     if rb=-1 then return(0)
2160   } else {
2170     if c1=c2 then {
2180       c_cnt=c_cnt+1
2190       oto(wake):wait(5)
2200       return(1)
2210     }
2220   }
2230   kake=0:rb=-1
2240   return(0)
2250 endfunc
2260 /*
2270 func check(no;int)          /* カードのチェック
2280   no=no mod 13
2290   if no=0 then no=13
2300   if no=1 then no=14
2310   if c_cnt=51 then c_cnt=rnd()*52
2320   return(no)
2330 endfunc
2340 /*
2350 func over()                 /* ゲーム終了？
2360   fill(88,256,423,351,8)
2370   int i
2380   apage(0)
2390   if kin>=10000 then win()
2400   s_hyouji(5,0,10)
2410   click()
2420   if rb=-1 then {
2430     cls
2440     apage(0):wipe():apage(1):wipe()
2450     color 3
2460     mouse(0)
2470     end
2480   }
2490   kin=100:m_hyouji()
2500   round=0
2510   s_hyouji(0,0,0)
2520   for i=0 to 4
2530     locate 23,18+i:print space$(18)
2540   next
2550   wipe():apage(1)
2560 endfunc
2570 /*
2580 func win()                   /* 勝ち
2590   fill(152,232,359,383,4)
2600   symbol(166,244,"CONGRATULATION!",1,1,2,15,0)
2610   locate 23,18:print"ラウンド数"
2620   locate 38,19:print using"###";round
2630   locate 23,21:print"最終獲得額"
2640   locate 34,22:print using"$######";kin
2650 endfunc
2660 /*
2670 func click()                 /* ボタンを押す
2680   repeat:msstat(x,y,lb,rb):until lb+rb<>0
2690 endfunc
2700 /*
2710 func c_hyouji(ptn;int)      /* カードの表示
2720   int i
2730   for i=0 to 4
2740     if ptn=0 then {
2750       c_put(88+i*72,256,0)
2760     } else {
2770       c_put(88+i*72,256,c(i))
2780     }
2790     oto(pasa):wait(2)
2800   next
2810 endfunc
2820 /*
2830 func y_hyouji(no;int,ptn;int)  /* 役の表示
2840   if ptn then color 15
2850   locate 28,no+1
2860   print using " &                               &   $###### ";yak
u(no),ritsu(no)*kake
2870   color 7
2880 endfunc
2890 /*
2900 func s_hyouji(m1;int,m2;int,m3;int)  /* 説明の表示
2910   locate 15,26:print m(m1)
2920   locate 15,27:print m(m2)
2930   locate 15,28:print m(m3)
2940 endfunc
2950 /*
2960 func m_hyouji()                        /* 持ち金の表示
2970   locate 19,9:print using "######";kin
2980 endfunc
2990 /*
3000 func k_hyouji()                        /* 賭け金の表示
3010   locate 56,10:print using "######";kake
3020 endfunc
3030 /*
3040 func jyunbi()                          /* 次の準備
3050   int i
3060   c_cnt=5
3070   for i=0 to 5
3080     tmp(i,0)=0:tmp(i,1)=0
3090     cc(i)=0
3100   next
3110   for i=0 to 7
3120     y_hyouji(i,0)
3130   next
3140 endfunc
3150 /*
3160 func oto(ptn;str)                      /* 効果音
3170   m_init():m_trk(1,ptn):m_play()
3180 endfunc
3190 /*
3200 func wait(w;int)                       /* ウェイト
3210   int i
3220   for i=0 to w*100:next
3230 endfunc
3240 /*
3250 func init()                            /* 初期設定
3260   screen 1,1,1,1
3270   console ,,0
3280   apage(1):vpage(3)
3290   msarea(88,256,423,351)
3300   mouse(4):mouse(1)
3310   m_assign(1,1):m_alloc(1,100)
3320   color 7
3330   randomize(val(mid$(time$,4,2)+right$(time$,2)))
3340 endfunc
3350 /*
3360 func nv_init()                         /* 変数の初期設定
3370   for i=0 to 51
3380     c(i)=i+1
3390   next
3400   round=0:kin=100:kake=0:up=10:max=50
3410   m(0)="                              "
3420   m(1)="賭け金を決めて下さい。"
3430   m(2)="カードを選んで下さい。"
3440   m(3)="ＤＯＵＢＬＥ　ＵＰ"
3450   m(4)="　　　　　　　に挑戦しますか？"
3460   m(5)="もう一度、挑戦しますか？"
3470   m(6)="賭け金は倍になりました。"
3480   m(7)="左ボタン・賭け金を増やす"
3490   m(8)="右ボタン・決定"
3500   m(9)="左ボタン・ＨＯＬＤ、ＣＡＮＳＥＬ"
3510   m(10)="左ボタン・ＹＥＳ　右ボタン・ＮＯ"
3520   m(11)="左ボタン・ＢＩＧ"
3530   m(12)="右ボタン・ＳＭＡＬＬ"
3540 endfunc
3550 /*
3560 func shuffle()                 /* シャッフル
3570   int i,a,b,k
3580   for i=0 to 99
3590     a=rnd()*52:b=rnd()*52
3600     k=c(a):c(a)=c(b):c(b)=k
3610   next
3620 endfunc
3630 /*
3640 func gamen()                           /* 画面描画
3650   fill(0,0,511,511,3)
3660   fill(0,192,511,511,8)
3670   fill(112,408,399,471,0)
3680   box(114,412,397,467,15)
3690   box(219,8,507,151,15)
3700   box(219,151,507,183,15)
3710   box(5,8,214,112,15)
3720   box(5,120,214,183,15)
3730   symbol(23,26,"THE",2,1,2,0,0)
3740   symbol(21,24,"THE",2,1,2,13,0)
3750   symbol(23,50,"POKER",3,2,2,0,0)
3760   symbol(21,48,"POKER",3,2,2,13,0)
3770   locate 2,9:print "あなたの持ち金　$"
3780   m_hyouji()
3790   locate 29,10:print "賭け金"+space$(20)+"$      0"
3800 endfunc
```

▶授業で使うパソコンはすべてPC-9801なので，友人はほぼすべてPC-9801を買っている。意志を貫いてX68000を買ったのは僕ひとりである。後悔はしていないがプログラムなどの授業では課題のやり直しが許されない（家でやれない）のでつらいです。
長崎　洋(22)神奈川県

バグレポートとファイル関数

亀田 雅彦 Kameda Masahiko

INTEGRAL X1の解説も今回でひとまずおしまい。ご迷惑をかけてきたバグを今回で一気に追放しましょう。また，INTEGRAL 用の外部コマンドを作成したという方はぜひ投稿お願いします。

読者の皆さん，約半年間のご愛読誠にありがとうございました。なんと！　ネタ不足のため，今月をもちましてこの連載は終了させていただくことになりました。今後も単発の形式で，KAME-DOSのサポートをしていくつもりです。お楽しみにね(ひそかにバージョンアップしてたりして)。

＊

さて，それでは今月のお題目。

1)　現在までに発見されているバグ特集。

2)　DOSらしく入出力関数群。

という2つです。まずは1)から。

バクだよ〜ん

どうも，おさわがせしてすみません。Oh!X1990年12月号の「ごめんなさいのコーナー」に掲載されたものを含めて，4カ所ほどバグがありました。以下に，それぞれの症状と修正方法を示します。

●その1

12月号のバグです。

＊症状

COPYコマンドでバイナリファイル(Binファイル)を上書きした場合，旧ファイルのロードアドレス，実行アドレスが新ファイルのものに更新されずに残ってしまいます。

＊対策

お手持ちの「FDC.OBJ」を，12月号のアセンブルリストのマシン語データ部分のように書き換えてください。または今月の変更用リスト3のように書き換えてください。そして，

SAVE "FDC.OBJ",&HD000,&HECFF

としていままでと同じようにセーブします。マシン語データを入力する際には，アドレスに注意してください。

●その2

＊症状

ASCIIセーブされたプログラムファイルをCOPYして，そのCOPYしたファイルを今度はBASICでLOADしようとすると，うまくいかないことがあります。

＊対策

アセンブルリストはリスト1。リスト3の変更でもかまいません。方法は「その1」をご覧ください。

●その3

＊症状

X1フォーマットのサブディレクトリにおいて，ファイルが8個までしか表示されませんでした。ただしそれはturboBASICで作られたファイルの場合です。KAME-DOSでサブディレクトリ下にコピーした場合は，8個以上あっても表示します。またMS-DOSフォーマットの場合はまったく正常に動作します。

＊対策

アセンブルリストはリスト2。リスト3の変更でもかまいません。方法は「その1」をご覧ください。

●その4

＊症状

「その3」に関連して，Oh!X8月号に掲載された「MD.X1」にも不備な点がありました。これで作ったサブディレクトリにturboBASICでファイルを書き込むと，正常に動作しないことがあります。

＊対策

「MD.X1」の1380行を以下のように修正します。

1380 k=1：i0=256：i1=&HC0：d=&HFF：RETURN

↓

1380 k=16：i0=4096：i1=&HC0：d=&HFF：RETURN

「MD.X1」はOh!X1990年8月号95，96ページに掲載されています。

＊　　＊　　＊

こんなにバグばっかりで，本当になんとお詫びしてよいやら。さすがにもう出尽くしたとは思うんですが……。

入出力関数

「FDC.OBJ」内には，意外と有用なサブルーチンも含まれているのですが，使い勝手がよくありません。そこで，BASICからでも簡単に呼び出せるように，いくつかのUSR関数を作ってみました。種類は8個。

FOPEN　：&HC000
FEOF　：&HC003
FGETC　：&HC006
FPUTC　：&HC009
FCLOSE　：&HC00C
FREADS　：&HC00F
FWRITS　：&HC012
FREAD　：&HC015

それぞれ，ファイルをシーケンシャルに読み出すための関数です。(名前の由来はX68000のX-BASIC)

ただしこのプログラムは，いわばDOSコールの拡張ですので，この関数を使ったプログラムがDOSの外部コマンドである必要はありません。つまり「FDC.OBJ」とこの関数ルーチンさえメモリ上にあれば，普通のBASICプログラムからでもDOS機能が使えるのです。

だいたいのコンセプトはわかりましたね。それでは実際の使い方です。

使い方

「CLEAR &HCC00」を実行後，リスト4をマシン語入力ツールで入力してください。そして，

SAVEM "FUNC.OBJ",&HCC00,&HCFFF

としてセーブします。この関数を使うためには「FDC.OBJ」をロードして，初期化しておく必要があります。「INTEGRAL.X」をLOAD，RUNしてください。そして先ほどの「FUNC.OBJ」をLOADします。これで準備完了です。このあとで，「CO

MMAND.X1」その他，外部コマンドを起動すると「FUNC.OBJ」は破壊されます。そのときは，

CLEAR &HCC00
LOADM "FUNC.OBJ"

を実行してください。

マシン語の準備が完了したら，呼び出すためのBASICプログラムを作ります。DEFUSR0～7に上記のアドレスを定義して使うのが簡単でしょう。具体的にサンプルプログラムを見てみます。なお今月のプログラムはすべて，ノーマルX1(CZ-8FB01)でも動作します。

*

サンプルプログラム。以下のプログラムは，「FUNC.OBJ」「FDC.OBJ」が準備されている状況で，ASCIIファイルを対象に実行してください。BASICからダイレクトにRUNできます。対象ファイルに正規のEOFがないと，うまく動作しないことがあります。

●リスト5（TYPE.BAS）

「FREADS」のサンプル。ファイルの中身を表示します。コマンドラインからTYPEするのとほぼ同じです。ファイル名は，プログラム中の1170行の"Y：COPY.DOC"で指定します。

プログラムは，ファイルOPENに始まり，EOFかエラー(PEEK(v_stop)<>0でエラー有）になるまで，改行コードごとに表示し続けます。読み込みの場合はCLOSEはいりません。

●リスト6（COPY.BAS）

「FGETC」「FPUTC」のサンプル。ファイルのコピーをします。1170行，1180行に，それぞれ読み込み/書き込みファイル名を入れて実行してください。1文字ずつ読み込み，改行以外のコントロールコードを抜かして書き込みます(表示もする)。

リスト3の使い方

まず「FDC.OBJ」をLOADします。

CLEAR &HD000
LOADM "FDC.OBJ"

次にマシン語入力ツールなどを使ってリスト3を入力します(もちろん「FDC.OBJ」をメモリ上においたまま)。そして，いままでの「FDC.OBJ」の代わりに，変更済みの「FDC.OBJ」をセーブします。

SAVEM "FDC.OBJ", &HD000, &HECFF

これで，12月号と今月号の分のすべてのバグを直したことになります。

リスト3　デバッグ部ダンプリスト

```
D565 C3 30 DE                      : D1
-----------------------------------
SUM: C3 30 DE 00 00 00 00 00 D9F1


DB15 CD 50 DE                      : FB
-----------------------------------
SUM: CD 50 DE 00 00 00 00 00 583F


D7E5 C2 71 DE                      : 11
-----------------------------------
SUM: C2 71 DE 00 00 00 00 00 88D0


E3EA CD 86 DE                      : 31
-----------------------------------
SUM: CD 86 DE 00 00 00 00 00 8E3F


E466 CD 86 DE                      : 31
-----------------------------------
SUM: CD 86 DE 00 00 00 00 00 8E3F


DE30 C2 42 E0 21 C8 D0 11 7E : 2C
DE38 E0 01 0D 00 ED B0 C3 68 : B6
DE40 D5 00 00 00 00 00 00 00 : D5
DE48 00 00 00 00 00 00 00 00 : 00
DE50 D9 C5 D5 79 B7 28 08 AF : 82
DE58 CD 18 E0 13 0D 18 F4 D1 : C2
DE60 C1 D9 C3 22 DB 00 00 00 : 5A
DE68 00 00 00 00 00 00 00 00 : 00
DE70 01 3A A1 E0 B7 C2 09 E0 : 1E
DE78 3E 01 32 70 DE CD 09 E0 : 75
DE80 3E 00 32 70 DE C9 F5 3A : B6
DE88 70 DE B7 28 0B F1 AF 32 : 0A
DE90 70 DE 3E 10 32 8F EB C9 : 11
DE98 F1 32 8F EB C9          : 66
-----------------------------------
SUM: 2C 22 EE B2 CD 98 71 5B 0FFD
```

リスト1　デバッグその1

```
0000              1  ;
0000              2  ;KAME-DOS BUG-FIX2 '90/11/10
0000              3  ;
0000              4  ;  S-OS REDA
0000              5
E018 P            6  #LDDEA  EQU $E018
DB22 P            7  #LDMSBT EQU $DB22
0000              8
DB15              9   ORG $DB15
DB15             10  ;
DB15 CD 50 DE    11   CALL BEGIN
DB18             12
DB18             13
DE50             14   ORG $DE50
DE50             15  ;
DE50             16  BEGIN
DE50 D9          17   EXX
DE51 C5          18   PUSH BC
DE52 D5          19   PUSH DE
DE53             20  BGLOOP
DE53 79          21     LD A,C
DE54 B7          22     OR A
DE55 28 08       23     JR Z,BGSKIP
DE57 AF          24     XOR A
DE58 CD 18 E0    25     CALL #LDDEA
DE5B 13          26     INC DE
DE5C 0D          27     DEC C
DE5D 18 F4       28     JR BGLOOP
DE5F             29  BGSKIP
DE5F D1          30   POP DE
DE60 C1          31   POP BC
DE61 D9          32   EXX
DE62 C3 22 DB    33   JP #LDMSBT
DE65             34
```

リスト2　デバッグその2

```
0000              1  ;
0000              2  ;KAME-DOS BUG-FIX3 '90/11/20
0000              3  ;
0000              4  ;  S-OS REDA
0000              5
E009 P            6  #CRSRW   EQU $E009
E0A1 P            7  #FRWF    EQU $E0A1
EB8F P            8  #EDD     EQU $EB8F
0000              9
D7E5             10   ORG $D7E5
D7E5             11  ;
D7E5 C2 71 DE    12   JP NZ,BEGIN1
D7E8             13
D7E8             14
DE70             15   ORG $DE70
DE70             16  ;
DE70             17  WORKS
DE70 01          18   DB 1
DE71             19
DE71             20  BEGIN1
DE71 3A A1 E0    21   LD A,(#FRWF)
DE74 B7          22   OR A
DE75 C2 09 E0    23   JP NZ,#CRSRW
DE78 3E 01       24   LD A,1
DE7A 32 70 DE    25   LD (WORKS),A
DE7D CD 09 E0    26   CALL #CRSRW
DE80 3E 00       27   LD A,0
DE82 32 70 DE    28   LD (WORKS),A
DE85 C9          29   RET
DE86             30
DE86             31  ;
DE86             32  BEGIN2
DE86 F5          33   PUSH AF
DE87 3A 70 DE    34   LD A,(WORKS)
DE8A B7          35   OR A
DE8B 28 0B       36   JR Z,SBG3
DE8D F1          37   POP AF
DE8E AF          38   XOR A
DE8F 32 70 DE    39   LD (WORKS),A
DE92 3E 10       40   LD A,$10
DE94 32 8F EB    41   LD (#EDD),A
DE97 C9          42   RET
DE98             43
DE98             44  SBG3
DE98 F1          45   POP AF
DE99 32 8F EB    46   LD (#EDD),A
DE9C C9          47   RET
DE9D             48
DE9D             49  ;
E3EA             50   ORG $E3EA
E3EA             51  ;
E3EA CD 86 DE    52   CALL BEGIN2
E3ED             53
E466             54   ORG $E466
E466             55  ;
E466 CD 86 DE    56   CALL BEGIN2
E469             57
```

そのため，かなり遅いので注意しましょう。

●リスト7(CON.BAS)

「WRITS」のサンプル。キーボートから1文字ずつ入力して，ファイルに書き出します。コントロールコードもそのまま書き込みます。CTRL+Zを入力した時点で，ファイルのCLOSEをして終了です。CLOSEしなければファイルは作成されません。

●リスト8(FDUMP.BAS)

「FREAD」のサンプル。指定ファイルを16進ダンプリスト形式で表示します。プログラムは，ファイルを16バイトごとに読み込んで表示するだけの簡単なものです。

＊　＊　＊

いずれのサンプルでも，指定ファイルがない，ディスクがささってないなどのエラーはF1$，F2$でチェックします。

関数リスト

各々の関数の引数，戻り値をリストアップします。なお，引数とは「USR(　)」の括弧内の値であり，戻り値は「A$=USR(　)」というときにA$に入る値です。引数，戻り値ともにすべて文字変数に統一されています。マシン語からコールする場合は，DEとBレジスタだけUSR関数形式にしておいてください。

●FOPEN

引数：CHR$(1)+“ファイル名”
　　　CHR$(2)+“ファイル名”
戻り値：ファイル番号

引数のCHR$(　)の中の数字は，1がREAD OPEN，2がWRITE OPENの指定です。ファイル名はフルパスで指定してください。

戻り値は「1か2か255」しかありません。実はKAME-DOSでは，同時に開けるファイル数はREADひとつとWRITEひとつの計2つだけなのです。したがって，READファイルのファイル番号は1，WRITEファイルは2と決まっています。戻り値が255の場合はエラーです。

●FEOF

引数：ファイル番号
戻り値：２５５か０

ファイル番号はわかりきっていますが，必ず指定してください。以下の関数も同様です。戻り値が255のときはEOF（エンドオブファイル）です。0のときは，まだファイルに続きがあります。

●FGETC

引数：ファイル番号
戻り値：データ

データというのは「ファイルのデータ1バイト」ということです。今回の関数群は，内部に共通のファイルポインタを持っていて，それに従ってデータを読み出してきます。FGETCの実行後は自動的にポインタが＋1されます。ポインタを任意で動かすFSEEKのような関数はありません。

エラーの場合はV_STOP(&HE08C)が0以外の数字になります。

●FPUTC

引数：データ
戻り値：不定

引数の1バイトデータをファイルに書き出します。実行後，自動的にポインタは－1されます。ほかはFGETCと同様です。

●FCLOSE

引数：ファイル番号
戻り値：不定

WRITEファイルをSAVEします。書き込み時は必ず最後にCLOSEしてください。逆にREADのときは必要ありません。

●FREADS

引数：ファイル番号
戻り値：読み込んだデータ数
　　　　データ(&HEE00～)

ファイルポインタから，次の改行まで読み込みます。戻り値はデータ数で，実際のデータは&HEE00～に格納されています。改行がみつからずに，「EOF・エラー・255バイトを超えた」場合はそこまでのデータが格納されています。データに改行コードは含まれません。

●FWRITS

引数：データ
戻り値：不定

FPUTCを複数バイト対応にしたものです。255バイト以下のデータを文字変数に入れて，引数としてください。コントロールコードも含めて，変数の長さ分だけは必ず書き込みます。

●FREAD

引数：読み込むデータ数
戻り値：読み込んだデータ数
　　　　データ(&HEE00～)

ファイルポインタから，指定された数だけデータを読み込みます。実際のデータは&HEE00～です。データ中のコントロールコードもまとめて読み出します。指定数に達しないうちに「EOF・エラー」になった場合は，そこまでのデータが入っています。

＊

今月のプログラムは，読者の皆さんが自分でプログラミングするときの役に立つように作りました。ファイル操作などで，MS-DOSフォーマットを扱いたいときなど，今月の関数を使ってみてください。BASICに限らず，マシン語からでもコールできます。

それでは，このDOSがより広く使われることを願ってこの連載を終わることとしましょう。

リスト4　FUNC.OBJ

```
CC00 C3 1C CC C3 F7 CC C3 2B : 1F
CC08 CD C3 B1 CD C3 32 CE C3 : 94
CC10 99 CE C3 FE CE C3 1B CF : A3
CC18 00 00 00 00 ED 53 62 CF : 71
CC20 3E 10 32 E5 EC 1A 13 05 : 83
CC28 FE 01 28 0C FE 02 28 5B : B6
CC30 3E FF ED 5B 62 CF 12 C9 : 91
CC38 3E 07 32 8B D1 CD E6 CC : 52
CC40 AF 32 81 D1 3E 01 32 80 : 24
CC48 D1 32 65 E0 C5 D5 CD 21 : D0
CC50 D0 D1 C1 3A 8C E0 B7 20 : DF
CC58 0F 68 3A 84 D1 47 7D 90 : 5A
CC60 6F 26 00 19 EB CD 24 D0 : 5A
CC68 3A 8C E0 B7 3E FF 20 15 : CF
CC70 AF 32 61 CF 32 65 CF 32 : A9
CC78 66 CF 21 00 00 22 18 CC : 5C
-----------------------------------
SUM: FE 14 FC 73 4D 1C 9F B5 3F0A

CC80 22 5D CF 3E 01 ED 5B 62 : 37
CC88 CF 12 C9 3E 07 32 8C D1 : 7E
CC90 CD E6 CC 3E 01 32 65 E0 : 35
CC98 3E 02 32 80 D1 3E 03 32 : 36
CCA0 81 D1 C5 D5 CD 21 D0 D1 : 7B
CCA8 C1 3A 8C E0 B7 20 1E 68 : C4
CCB0 3A 84 D1 47 7D 90 6F 26 : 78
CCB8 00 19 EB 3E 01 32 65 E0 : BA
CCC0 3E 02 32 80 D1 3E 03 32 : 36
CCC8 81 D1 CD 24 D0 3A 8C E0 : B9
CCD0 B7 3E FF 20 0B 21 00 00 : 40
CCD8 22 1A CC 22 5F CF 3E 02 : 98
CCE0 ED 5B 62 CF 12 C9 C5 3A : 53
CCE8 8B E0 4F 06 00 21 C0 EC : 8D
CCF0 09 7E 32 97 E0 C1 C9 F5 : AF
CCF8 C5 D5 E5 D5 11 64 CF CD : 65
-----------------------------------
SUM: 56 B8 35 9B EA 09 FB 80 5D2B

CD00 06 CC 2A 18 CC 2B 22 18 : 45
CD08 CC 2A 5D CF 23 22 5D CF : 93
CD10 D1 AF 12 32 65 CF 3A 64 : 96
CD18 CF 47 3A E7 EC B8 20 06 : 01
CD20 3E FF 12 32 65 CF E1 D1 : 67
CD28 C1 F1 C9 F5 C5 D5 E5 DD : CC
CD30 E5 3A 65 CF B7 20 3E ED : 55
CD38 53 62 CF 2A 5D CF 7C B5 : 0B
CD40 CC 7C CD 3A 8C E0 B7 20 : 92
CD48 25 2A 18 CC ED 5B E3 EC : 4A
CD50 CD 2A E0 ED 5B DF EC 19 : 03
CD58 CD 12 E0 ED 5B 62 CF 12 : 4A
CD60 2A 18 CC 23 22 18 CC 2A : 61
CD68 5D CF 2B 22 5D CF DD E1 : 63
CD70 E1 D1 C1 F1 C9 3E 09 32 : A6
CD78 8C E0 18 F2 3A 66 CF B7 : 9C
-----------------------------------
SUM: 28 F2 57 28 2F 6E 2F CC 3E13

CD80 28 06 3E 09 32 8C E0 C9 : DC
CD88 3E 07 32 8B D1 3E 01 32 : 44
CD90 8C D1 3A 61 CF 32 9C E0 : 75
CD98 CD 0C D0 3A 9C E0 B7 20 : 36
CDA0 06 3E FF 32 66 CF AF 32 : 8B
CDA8 61 CF 2A 9D E0 22 5D CF : 25
CDB0 C9 F5 C5 D5 E5 DD E5 78 : 77
CDB8 B7 28 2E 2A DF EC ED 4B : 3A
CDC0 E3 EC 09 ED 4B 5F CF 09 : 47
CDC8 1A EB CD 18 E0 2A 1A CC : DA
CDD0 23 22 1A CC 2A 5F CF 23 : A6
CDD8 22 5F CF AF 32 9B E0 ED : 99
CDE0 5B E3 EC B7 ED 52 CC F0 : DC
CDE8 CD DD E1 E1 D1 C1 F1 C9 : B8
CDF0 2A 5F CF 22 9D E0 3E 01 : 36
CDF8 32 8B D1 3E 07 32 8C D1 : 62
-----------------------------------
SUM: 6C 16 C2 75 61 3E 31 2F B162

CE00 3E 02 32 9C E0 CD 0F CE : 98
CE08 21 00 00 22 5F CF C9 3E : 78
CE10 CD 21 29 CE 32 7A DB 22 : 8E
CE18 7B DB CD 0C D0 3E 2A 21 : 88
CE20 A4 E0 32 7A DB 22 7B DB : 83
CE28 C9 2A DF EC ED 5B E3 EC : D5
CE30 19 C9 F5 C5 D5 E5 DD E5 : 18
CE38 1A FE 02 20 39 11 64 CF : B7
CE40 3A E8 EC 12 06 01 CD 09 : FD
CE48 CC CD 7D CE 3E 01 32 9B : F0
CE50 E0 CD F0 CD 2A 1A CC 22 : 9C
```

```
CE58 86 D1 21 00 00 22 88 D1 : F3
CE60 3A 08 D1 E6 01 3E 04 20 : 5C
CE68 02 3E 20 32 82 D0 3E 02 : 24
CE70 32 80 D1 CD 15 D0 DD E1 : F3
CE78 E1 D1 C1 F1 C9 2A DF EC : 22
---------------------------------
SUM: 02 B9 2D 66 E6 0D CD 50 FD90

CE80 ED 5B E3 EC 19 ED 5B 5F : D7
CE88 CF 19 EB AF CD 18 E0 13 : 5A
CE90 2A E3 EC B7 ED 52 20 F3 : 02
CE98 C9 F5 C5 D5 E5 DD E5 D5 : D4
CEA0 06 FF 21 00 EE 11 64 CF : 58
CEA8 3E 01 12 CD 06 CC 1A 4F : 59
```

```
CEB0 3A E7 EC B9 20 10 2A 18 : 38
CEB8 CC 2B 22 18 CC 2A 5D CF : 53
CEC0 23 22 5D CF 18 2D 79 FE : 2D
CEC8 0D 20 1E 11 64 CF 3E 01 : CE
CED0 12 CD 06 CC 1A FE 0A 28 : FB
CED8 1A 2A 18 CC 2B 22 18 CC : 59
CEE0 2A 5D CF 23 22 5D CF 18 : DF
CEE8 0A 3A 8C E0 B7 20 04 71 : FC
CEF0 23 10 B2 78 2F D1 12 DD : 4C
CEF8 E1 E1 D1 C1 F1 C9 F5 C5 : C8
---------------------------------
SUM: 8D 1F 37 79 52 7E F8 5D 2ED7

CF00 D5 E5 DD E5 78 B7 28 0C : DF
```

```
CF08 CD 09 CC 3A 8C E0 B7 20 : 1F
CF10 03 13 10 F4 DD E1 E1 D1 : 8A
CF18 C1 F1 C9 F5 C5 D5 E5 DD : CC
CF20 E5 D5 1A 47 21 00 EE 11 : 3B
CF28 64 CF 3E 01 12 CD 06 CC : 23
CF30 1A 4F 3A E7 EC B9 20 10 : 5F
CF38 2A 18 CC 2B 22 18 CC 2A : 69
CF40 5D CF 23 22 5D CF 18 0A : BF
CF48 3A 8C E0 B7 20 04 71 23 : 15
CF50 10 D5 D1 1A 90 12 DD E1 : 30
CF58 E1 D1 C1 F1 C9 00 00 00 : 2D
CF60 00 00 00 00 00 00 00    : 00
---------------------------------
SUM: 7B FE 75 46 BD D0 EB FF 510B
```

リスト5 TYPE.BAS

```
1000 '
1010 'TYPE.BAS            FOR INTEGRAL X (WITH FDC.OBJ)
1020 '                                       By M.Kameda
1030 CLEAR &HCC00
1040 'LOADM "1:FUNC.OBJ"
1050 DEFINT A-Z:v_stop=&HE08C:p256=&HEE00
1060 '                   *NAME                            *RETURN
1070 DEFUSR0=&HCC00    'FOPEN  USR(R(1)/W(2) , FILE NAME) FILE NO.
1080 DEFUSR1=&HCC03    'FEOF   USR(FILE NO.)              EOF(-1) NOT(0)
1090 DEFUSR2=&HCC06    'FGETC  USR(FILE NO.)              DATA
1100 DEFUSR3=&HCC09    'FPUTC  USR(DATA)                   ?
1110 DEFUSR4=&HCC0C    'FCLOSE USR(FILE NO.)               ?
1120 DEFUSR5=&HCC0F    'FREADS USR(FILE NO.)              LENGTH , p256-
1130 DEFUSR6=&HCC12    'FWRITS USR(DATA)                   ?
1140 DEFUSR7=&HCC15    'FREAD  USR(LENGTH)                LENGTH , p256-
1150 '                   FILE NO.=1(READ)/2(WRITE)
1160 '
1170 D1$=USR0(CHR$(1)+"Y:COPY.DOC") '自分のファイル名にする  'OPEN
1180 '
1190 F1$=LEFT$(D1$,1):IF ASC(F1$)<>1 THEN 1290
1200 '
1210 E$=USR1(F1$+"")
1220 WHILE ASC(E$)=0 AND PEEK(v_stop)=0                        'EOF?
1230   W$=USR5(F1$+""):L=ASC(W$)                               'FREADS
1240   IF PEEK(v_stop)=0 THEN PRINT MEM$(p256,L)
1250   E$=USR1(F1$+"")
1260 WEND
1270 END
1280 '
1290 PRINT "ERROR!":STOP
```

リスト6 COPY.BAS

```
1000 '
1010 'COPY.BAS            FOR INTEGRAL X (WITH FDC.OBJ)
1020 '                                       By M.Kameda
1030 CLEAR &HCC00
1040 'LOADM "1:FUNC.OBJ"
1050 DEFINT A-Z:v_stop=&HE08C:p256=&HEE00
1060 '                   *NAME                            *RETURN
1070 DEFUSR0=&HCC00    'FOPEN  USR(R(1)/W(2) , FILE NAME) FILE NO.
1080 DEFUSR1=&HCC03    'FEOF   USR(FILE NO.)              EOF(-1) NOT(0)
1090 DEFUSR2=&HCC06    'FGETC  USR(FILE NO.)              DATA
1100 DEFUSR3=&HCC09    'FPUTC  USR(DATA)                   ?
1110 DEFUSR4=&HCC0C    'FCLOSE USR(FILE NO.)               ?
1120 DEFUSR5=&HCC0F    'FREADS USR(FILE NO.)              LENGTH , p256-
1130 DEFUSR6=&HCC12    'FWRITS USR(DATA)                   ?
1140 DEFUSR7=&HCC15    'FREAD  USR(LENGTH)                LENGTH , p256-
1150 '                   FILE NO.=1(READ)/2(WRITE)
1160 '
1170 D1$=USR0(CHR$(1)+"W:DOS3.RED") '自分のファイル名にする  'OPEN READ
1180 D2$=USR0(CHR$(2)+"Y:COPY.DOC") '自分のファイル名にする  'OPEN WRITE
1190 '
1200 F1$=LEFT$(D1$,1):IF ASC(F1$)<>1 THEN 1330
1210 F2$=LEFT$(D2$,1):IF ASC(F2$)<>2 THEN 1330
1220 '
1230 E$=USR1(F1$+"")
1240 WHILE ASC(E$)=0 AND PEEK(v_stop)=0                        'EOF?
1250   D$=USR2(F1$+"")                                         'FGETC
1260   IF ASC(D$)<32 AND ASC(D$)<>13 THEN 1290    '改行以外のCTRL-CODE
1270   IF PEEK(v_stop)=0 THEN PRINT D$;:W$=USR3(D$+"")         'FPUTC
1280   E$=USR1(F1$+"")
1290 WEND
1300 W$=USR4(F2$+"")                                           'FCLOSE
1310 END
1320 '
1330 PRINT "ERROR!":STOP
```

リスト7 CON.BAS

```
1000 '
1010 'CON.BAS             FOR INTEGRAL X (WITH FDC.OBJ)
1020 '                                       By M.Kameda
1030 CLEAR &HCC00
1040 'LOADM "1:FUNC.OBJ"
1050 DEFINT A-Z:v_stop=&HE08C:v_mac=&HE097:s_eof=&HECE7
1060 '                   *NAME                            *RETURN
1070 DEFUSR0=&HCC00    'FOPEN  USR(R(1)/W(2) , FILE NAME) FILE NO.
1080 DEFUSR1=&HCC03    'FEOF   USR(FILE NO.)              EOF(-1) NOT(0)
1090 DEFUSR2=&HCC06    'FGETC  USR(FILE NO.)              DATA
1100 DEFUSR3=&HCC09    'FPUTC  USR(DATA)                   ?
1110 DEFUSR4=&HCC0C    'FCLOSE USR(FILE NO.)               ?
1120 DEFUSR5=&HCC0F    'FREADS USR(FILE NO.)              LENGTH , p256-
1130 DEFUSR6=&HCC12    'FWRITS USR(DATA)                   ?
1140 DEFUSR7=&HCC15    'FREAD  USR(LENGTH)                LENGTH , p256-
1150 '                   FILE NO.=1(READ)/2(WRITE)
```

```
1160 '
1170 D2$=USR0(CHR$(2)+"Y:COPY.DOC") '自分のファイル名にする  'OPEN WRITE
1180 M=PEEK(v_mac):IF M=2 OR M=4 THEN CR$=CHR$(13,10) ELSE CR$=CHR$(13)
1190 '                                     MS-DOS                  X1
1200 F2$=LEFT$(D2$,1):IF ASC(F2$)<>2 THEN 1350
1210 '
1220 REPEAT
1230   S$="":D$=""
1240   REPEAT                                                      '
1250     S$=S$+D$                                                  '
1260     D$=INKEY$(1):D=ASC(D$)                                    '1行入力
1270     PRINT D$;                                                 '13..[RET]
1280   UNTIL D=26 OR D=13                                          '26..CTRL+Z
1290   IF D=13 THEN S$=S$+CR$
1300   W$=USR6(S$)                                                 'FWRITS
1310 UNTIL D=26
1320 W$=USR4(F2$+"")                                               'FCLOSE
1330 END
1340 '
1350 PRINT "ERROR!":STOP
```

リスト8 FDUMP.BAS

```
1000 '
1010 'FDUMP.BAS              FOR INTEGRAL X (WITH FDC.OBJ)
1020 '                                             By M.Kameda
1030 CLEAR &HCC00
1040 'LOADM "1:FUNC.OBJ"
1050 DEFINT A-Z:v_stop=&HE08C:p256=&HEE00
1060 '                   *NAME                                      *RETURN
1070 DEFUSR0=&HCC00    'FOPEN  USR(R(1)/W(2) , FILE NAME) FILE NO.
1080 DEFUSR1=&HCC03    'FEOF   USR(FILE NO.)              EOF(-1) NOT(0)
1090 DEFUSR2=&HCC06    'FGETC  USR(FILE NO.)              DATA
1100 DEFUSR3=&HCC09    'FPUTC  USR(DATA)                   ?
1110 DEFUSR4=&HCC0C    'FCLOSE USR(FILE NO.)               ?
1120 DEFUSR5=&HCC0F    'FREADS USR(FILE NO.)              LENGTH , p256-
1130 DEFUSR6=&HCC12    'FWRITS USR(DATA)                   ?
1140 DEFUSR7=&HCC15    'FREAD  USR(LENGTH)                LENGTH , p256-
1150 '                    FILE NO.=1(READ)/2(WRITE)
1160 '
1170 D1$=USR0(CHR$(1)+"Y:COPY.DOC") '自分のファイル名にする 'OPEN
1180 '
1190 F1$=LEFT$(D1$,1):IF ASC(F1$)<>1 THEN 1320
1200 '
1210 E$=USR1(F1$+"")
1220 WHILE ASC(E$)=0 AND PEEK(v_stop)=0                          'EOF?
1230   W$=USR7(CHR$(16)):L=ASC(W$)                               'FREADS
1240   FOR I=0 TO L-1
1250     PRINT RIGHT$("0"+HEX$(ASC(MEM$(p256+I,1))),2);" ";
1260   NEXT
1270   PRINT TAB(49);:PRINT#0"/  ";MEM$(p256,L)
1280   E$=USR1(F1$+"")
1290 WEND
1300 END
1310 '
1320 PRINT "ERROR!":STOP
```

リスト9 FUNC.RED

```
0000                  1  ;
0000                  2  ;FUNC.RED
0000                  3  ;        WITH KAME-DOS
0000                  4  ;                   Ver 1.0
0000                  5  ;     FOR S-OS REDA
0000                  6
CC00                  7   ORG $CC00
CC00                  8
D080 P                9  #ZOKU    EQU $D080 ;POKE ADR.
D180 P               10  #OD      EQU $D180
D181 P               11  #OP      EQU $D181
D182 P               12  #FCRS    EQU $D182
D184 P               13  #YEN     EQU $D184
D186 P               14  #FSZL    EQU $D186
D18A P               15  #DDRV    EQU $D18A
E054 P               16  #FLNM    EQU $E054
E065 P               17  #SBDR    EQU $E065
E068 P               18  #FNAM    EQU $E068
E078 P               19  #FBYT    EQU $E078
E08C P               20  #STOP    EQU $E08C
E08B P               21  #DN      EQU $E08B
E091 P               22  #BF      EQU $E091
E097 P               23  #MAC     EQU $E097
E09B P               24  #EDR     EQU $E09B
E09C P               25  #IOFG    EQU $E09C
E09D P               26  #MSBT    EQU $E09D
ECC0 P               27  &MAC4    EQU $ECC0
ECE3 P               28  &BSIZ    EQU $ECC0+35
ECE5 P               29  &ESCP    EQU $ECC0+37
ECE7 P               30  &EOF     EQU $ECC0+39
ECE8 P               31  &EOF3    EQU $ECC0+40
ECDF P               32  #SBUFF   EQU $ECDF
EE00 P               33  P256     EQU $EE00
CC00                 34
E003 P               35  #DLFAT   EQU $E003 ;CALL ADR.
E012 P               36  #LDAHL   EQU $E012
E015 P               37  #LDADE   EQU $E015
E018 P               38  #LDDEA   EQU $E018
E02A P               39  #DIVHD   EQU $E02A
D006 P               40  #DIR2    EQU $D006
D00C P               41  #DEVI    EQU $D00C
D012 P               42  #DLDIR   EQU $D012
D015 P               43  #SAVED   EQU $D015
D021 P               44  #PREOP   EQU $D021
D024 P               45  #OPENS   EQU $D024
D02A P               46  #DIRSB   EQU $D02A
D108 P               47  #FMAC    EQU $D108
2F81 P               48  SFTJIS@  EQU $2F81
2FB6 P               49  JISVRM@  EQU $2FB6
3099 P               50  SFTCHK@  EQU $3099
CC00                 51
CC00                 52  ;
CC00 C3 1C CC        53  FOPEN    JP OPEN1   ;USR
CC03 C3 F7 CC        54  FEOFJ    JP EOF?    ;USR
CC06 C3 2B CD        55  FGETC    JP GETC    ;USR
CC09 C3 B1 CD        56  FPUTC    JP PUTC    ;USR
CC0C C3 32 CE        57  FCLOSE   JP CLOSE   ;USR
CC0F C3 99 CE        58  FREADS   JP READS   ;USR
CC12 C3 FE CE        59  FWRITS   JP WRITS   ;USR
CC15 C3 1B CF        60  FREAD    JP READ    ;USR
CC18                 61  ;
CC18 00 00           62  FPOT1   DW 0       ;R
CC1A 00 00           63  FPOT2   DW 0       ;R
CC1C                 64  ;
CC1C                 65
CC1C                 66  ;
CC1C                 67  ;FOPEN
CC1C                 68  ; (R,W)-(FILENAME)
CC1C                 69  ;
CC1C                 70  OPEN1
CC1C ED 53 62 CF     71   LD (USRWK),DE
CC20 3E 10           72   LD A,$10
CC22 32 E5 EC        73   LD (&ESCP),A
CC25 1A              74   LD A,(DE)
CC26 13              75   INC DE
CC27 05              76   DEC B
CC28 FE 01           77   CP 1
CC2A 28 0C           78   JR Z,OPENR
CC2C FE 02           79   CP 2
CC2E 28 5B           80   JR Z,OPENW
CC30 3E FF           81   LD A,$FF
CC32 ED 5B 62 CF     82   LD DE,(USRWK)
CC36 12              83   LD (DE),A
CC37 C9              84   RET
CC38                 85
CC38                 86  OPENR
CC38 3E 07           87   LD A,7
CC3A 32 8B D1        88   LD (#DDRV+1),A
CC3D CD E6 CC        89   CALL SMACDN
CC40 AF              90   XOR A
CC41 32 81 D1        91   LD (#OP),A
CC44 3E 01           92   LD A,1
CC46 32 80 D1        93   LD (#OD),A
CC49 32 65 E0        94   LD (#SBDR),A
CC4C C5              95   PUSH BC
CC4D D5              96   PUSH DE
CC4E CD 21 D0        97   CALL #PREOP
CC51 D1              98   POP DE
CC52 C1              99   POP BC
CC53 3A 8C E0       100   LD A,(#STOP)
CC56 B7             101   OR A
CC57 20 0F          102   JR NZ,RETVR
CC59 68             103   LD L,B
CC5A 3A 84 D1       104   LD A,(#YEN)
CC5D 47             105   LD B,A
CC5E 7D             106   LD A,L
CC5F 90             107   SUB B
CC60 6F             108   LD L,A
CC61 26 00          109   LD H,0
CC63 19             110   ADD HL,DE
CC64 EB             111   EX DE,HL
CC65 CD 24 D0       112   CALL #OPENS
CC68                113  RETVR
CC68 3A 8C E0       114   LD A,(#STOP)
CC6B B7             115   OR A
CC6C 3E FF          116   LD A,$FF
CC6E 20 15          117   JR NZ,ERRRETR
CC70 AF             118     XOR A
CC71 32 61 CF       119     LD (IOFG1),A
CC74 32 65 CF       120     LD (FEOF),A
CC77 32 66 CF       121     LD (EOR),A
CC7A 21 00 00       122     LD HL,0
CC7D 22 18 CC       123     LD (FPOT1),HL
CC80 22 5D CF       124     LD (INBP1),HL
CC83 3E 01          125     LD A,1
CC85                126  ERRRETR
CC85 ED 5B 62 CF    127   LD DE,(USRWK)
CC89 12             128   LD (DE),A
CC8A C9             129   RET
CC8B                130
CC8B                131  OPENW
CC8B 3E 07          132   LD A,7
CC8D 32 8C D1       133   LD (#DDRV+2),A
CC90 CD E6 CC       134   CALL SMACDN
CC93 3E 01          135   LD A,1
CC95 32 65 E0       136   LD (#SBDR),A
CC98 3E 02          137   LD A,2
CC9A 32 80 D1       138   LD (#OD),A
CC9D 3E 03          139   LD A,3
CC9F 32 81 D1       140   LD (#OP),A
CCA2 C5             141   PUSH BC
CCA3 D5             142   PUSH DE
CCA4 CD 21 D0       143   CALL #PREOP
CCA7 D1             144   POP DE
CCA8 C1             145   POP BC
CCA9 3A 8C E0       146   LD A,(#STOP)
CCAC B7             147   OR A
CCAD 20 1E          148   JR NZ,RETVW
CCAF 68             149   LD L,B
CCB0 3A 84 D1       150   LD A,(#YEN)
CCB3 47             151   LD B,A
CCB4 7D             152   LD A,L
CCB5 90             153   SUB B
CCB6 6F             154   LD L,A
CCB7 26 00          155   LD H,0
CCB9 19             156   ADD HL,DE
CCBA EB             157   EX DE,HL
CCBB 3E 01          158   LD A,1
CCBD 32 65 E0       159   LD (#SBDR),A
CCC0 3E 02          160   LD A,2
CCC2 32 80 D1       161   LD (#OD),A
CCC5 3E 03          162   LD A,3
CCC7 32 81 D1       163   LD (#OP),A
CCCA CD 24 D0       164   CALL #OPENS
CCCD                165  RETVW
CCCD 3A 8C E0       166   LD A,(#STOP)
CCD0 B7             167   OR A
CCD1 3E FF          168   LD A,$FF
CCD3 20 0B          169   JR NZ,ERRRETW
CCD5 21 00 00       170     LD HL,0
CCD8 22 1A CC       171     LD (FPOT2),HL
CCDB 22 5F CF       172     LD (INBP2),HL
CCDE 3E 02          173     LD A,2
CCE0                174  ERRRETW
CCE0 ED 5B 62 CF    175   LD DE,(USRWK)
CCE4 12             176   LD (DE),A
CCE5 C9             177   RET
CCE6                178
CCE6                179  ;
CCE6                180  SMACDN
```

▶受験生は各自の責任において解凍……。そうはいっても記事を読んだら解凍せずにはいられない。今年の冬は長くなるかもしれません。　川野　啓祐(18)千葉県

```
CCE6 C5          181  PUSH BC
CCE7 3A 8B E0    182  LD A,(#DN)
CCEA 4F          183  LD C,A
CCEB 06 00       184  LD B,0
CCED 21 C0 EC    185  LD HL,&MAC4
CCF0 09          186  ADD HL,BC
CCF1 7E          187  LD A,(HL)
CCF2 32 97 E0    188  LD (#MAC),A
CCF5 C1          189  POP BC
CCF6 C9          190  RET
CCF7             191
CCF7             192 ;
CCF7             193 ;FEOF()
CCF7             194 ; (FILNO)
CCF7             195 ;
CCF7             196 EOF?
CCF7 F5          197  PUSH AF
CCF8 C5          198  PUSH BC
CCF9 D5          199  PUSH DE
CCFA E5          200  PUSH HL
CCFB D5          201  PUSH DE
CCFC 11 64 CF    202  LD DE,EOFWK
CCFF CD 06 CC    203  CALL FGETC
CD02 2A 18 CC    204     LD HL,(FPOT1)
CD05 2B          205     DEC HL
CD06 22 18 CC    206     LD (FPOT1),HL
CD09 2A 5D CF    207     LD HL,(INBP1)
CD0C 23          208     INC HL
CD0D 22 5D CF    209     LD (INBP1),HL
CD10 D1          210  POP DE
CD11 AF          211  XOR A
CD12 12          212  LD (DE),A
CD13 32 65 CF    213  LD (FEOF),A
CD16 3A 64 CF    214  LD A,(EOFWK)
CD19 47          215  LD B,A
CD1A 3A E7 EC    216  LD A,(&EOF)
CD1D B8          217  CP B
CD1E 20 06       218  JR NZ,EOFRET
CD20 3E FF       219  LD A,-1
CD22 12          220  LD (DE),A
CD23 32 65 CF    221  LD (FEOF),A
CD26             222 EOFRET
CD26 E1          223  POP HL
CD27 D1          224  POP DE
CD28 C1          225  POP BC
CD29 F1          226  POP AF
CD2A C9          227  RET
CD2B             228
CD2B             229 ;
CD2B             230 ;FGETC [1BYTE FROM FILE]
CD2B             231 ; (FILNO)
CD2B             232 ;
CD2B             233 GETC
CD2B F5          234  PUSH AF
CD2C C5          235  PUSH BC
CD2D D5          236  PUSH DE
CD2E E5          237  PUSH HL
CD2F DD E5       238  PUSH IX
CD31 3A 65 CF    239  LD A,(FEOF)
CD34 B7          240  OR A
CD35 20 3E       241  JR NZ,EXCREAD
CD37 ED 53 62 CF 242  LD (USRWK),DE
CD3B 2A 5D CF    243  LD HL,(INBP1)
CD3E 7C          244  LD A,H
CD3F B5          245  OR L
CD40 CC 7C CD    246  CALL Z,GETDEIO
CD43 3A 8C E0    247     LD A,(#STOP)
CD46 B7          248     OR A
CD47 20 25       249     JR NZ,GETRET
CD49 2A 18 CC    250  LD HL,(FPOT1)
CD4C ED 5B E3 EC 251  LD DE,(&BSIZ)
CD50 CD 2A E0    252  CALL #DIVHD
CD53 ED 5B DF EC 253  LD DE,(#SBUFF)
CD57 19          254  ADD HL,DE
CD58 CD 12 E0    255  CALL #LDAHL
CD5B ED 5B 62 CF 256  LD DE,(USRWK)
CD5F 12          257  LD (DE),A
CD60             258
CD60 2A 18 CC    259  LD HL,(FPOT1)
CD63 23          260  INC HL
CD64 22 18 CC    261  LD (FPOT1),HL
CD67 2A 5D CF    262  LD HL,(INBP1)
CD6A 2B          263  DEC HL
CD6B 22 5D CF    264  LD (INBP1),HL
CD6E             265 GETRET
CD6E DD E1       266  POP IX
CD70 E1          267  POP HL
CD71 D1          268  POP DE
CD72 C1          269  POP BC
CD73 F1          270  POP AF
CD74 C9          271  RET
CD75             272
CD75             273 ;
CD75             274 EXCREAD
CD75 3E 09       275  LD A,9
CD77 32 8C E0    276  LD (#STOP),A
CD7A 18 F2       277  JR GETRET
CD7C             278
CD7C             279 ;
CD7C             280 GETDEIO
CD7C 3A 66 CF    281  LD A,(EOR)
CD7F B7          282  OR A
CD80 28 06       283  JR Z,GETSK1
CD82 3E 09       284     LD A,9
CD84 32 8C E0    285     LD (#STOP),A
CD87 C9          286     RET
CD88             287 GETSK1
CD88 3E 07       288  LD A,7
CD8A 32 8B D1    289  LD (#DDRV+1),A
CD8D 3E 01       290  LD A,1
CD8F 32 8C D1    291  LD (#DDRV+2),A
CD92 3A 61 CF    292  LD A,(IOFG1)
CD95 32 9C E0    293  LD (#IOFG),A
CD98 CD 0C D0    294  CALL #DEVI
CD9B 3A 9C E0    295  LD A,(#IOFG)
CD9E B7          296  OR A
CD9F 20 06       297  JR NZ,GETDESK
CDA1 3E FF       298     LD A,-1
CDA3 32 66 CF    299     LD (EOR),A
CDA6 AF          300     XOR A
CDA7             301 GETDESK
CDA7 32 61 CF    302  LD (IOFG1),A
CDAA 2A 9D E0    303  LD HL,(#MSBT)
CDAD 22 5D CF    304  LD (INBP1),HL
CDB0 C9          305  RET
CDB1             306
CDB1             307 ;
CDB1             308 ;FPUTC [1BYTE TO FILE]
CDB1             309 ; (DATA)
CDB1             310 ;
CDB1             311 PUTC
CDB1 F5          312  PUSH AF
CDB2 C5          313  PUSH BC
CDB3 D5          314  PUSH DE
CDB4 E5          315  PUSH HL
CDB5 DD E5       316  PUSH IX
CDB7 78          317  LD A,B
CDB8 B7          318  OR A
CDB9 28 2E       319  JR Z,PUTRET
CDBB 2A DF EC    320  LD HL,(#SBUFF)
CDBE ED 4B E3 EC 321  LD BC,(&BSIZ)
CDC2 09          322  ADD HL,BC
CDC3 ED 4B 5F CF 323  LD BC,(INBP2)
CDC7 09          324  ADD HL,BC
CDC8 1A          325  LD A,(DE)
CDC9 EB          326  EX DE,HL
CDCA CD 18 E0    327  CALL #LDDEA
CDCD 2A 1A CC    328  LD HL,(FPOT2)
CDD0 23          329  INC HL
CDD1 22 1A CC    330  LD (FPOT2),HL
CDD4 2A 5F CF    331  LD HL,(INBP2)
CDD7 23          332  INC HL
CDD8 22 5F CF    333  LD (INBP2),HL
CDDB AF          334  XOR A
CDDC 32 9B E0    335  LD (#EDR),A
CDDF ED 5B E3 EC 336  LD DE,(&BSIZ)
CDE3 B7          337  OR A
CDE4 ED 52       338  SBC HL,DE
CDE6 CC F0 CD    339  CALL Z,PUTDEIO
CDE9             340 PUTRET
CDE9 DD E1       341  POP IX
CDEB E1          342  POP HL
CDEC D1          343  POP DE
CDED C1          344  POP BC
CDEE F1          345  POP AF
CDEF C9          346  RET
CDF0             347
CDF0             348 ;
CDF0             349 PUTDEIO
CDF0 2A 5F CF    350  LD HL,(INBP2)
CDF3 22 9D E0    351  LD (#MSBT),HL
CDF6 3E 01       352  LD A,1
CDF8 32 8B D1    353  LD (#DDRV+1),A
CDFB 3E 07       354  LD A,7
CDFD 32 8C D1    355  LD (#DDRV+2),A
CE00 3E 02       356  LD A,2
CE02 32 9C E0    357  LD (#IOFG),A
CE05 CD 0F CE    358  CALL PUTDE2
CE08 21 00 00    359  LD HL,0
CE0B 22 5F CF    360  LD (INBP2),HL
CE0E C9          361  RET
CE0F             362
CE0F             363 ;
CE0F             364 PUTDE2
CE0F 3E CD       365     LD A,$CD         ;
CE11 21 29 CE    366     LD HL,PATCHPUT  ;CALL
CE14 32 7A DB    367     LD ($DB7A),A    ;PATCHPUT
CE17 22 7B DB    368     LD ($DB7B),HL   ;
CE1A CD 0C D0    369  CALL #DEVI
CE1D 3E 2A       370     LD A,$2A
CE1F 21 A4 E0    371     LD HL,$E0A4
CE22 32 7A DB    372     LD ($DB7A),A
CE25 22 7B DB    373     LD ($DB7B),HL
CE28 C9          374  RET
CE29             375
CE29             376 PATCHPUT
CE29 2A DF EC    377  LD HL,(#SBUFF)
CE2C ED 5B E3 EC 378  LD DE,(&BSIZ)
CE30 19          379  ADD HL,DE
CE31 C9          380  RET
CE32             381
CE32             382 ;
CE32             383 ;FCLOSE
CE32             384 ; (FILNO)
CE32             385 ;
CE32             386 CLOSE
CE32 F5          387  PUSH AF
CE33 C5          388  PUSH BC
CE34 D5          389  PUSH DE
CE35 E5          390  PUSH HL
CE36 DD E5       391  PUSH IX
CE38 1A          392  LD A,(DE)
CE39 FE 02       393  CP 2
CE3B 20 39       394  JR NZ,CLOSERET
CE3D 11 64 CF    395  LD DE,EOFWK
CE40 3A E8 EC    396  LD A,(&EOF3)
CE43 12          397  LD (DE),A
CE44 06 01       398  LD B,1
CE46 CD 09 CC    399  CALL FPUTC
CE49 CD 7D CE    400  CALL CLRBF
CE4C 3E 01       401  LD A,1
CE4E 32 9B E0    402  LD (#EDR),A
CE51 CD F0 CD    403  CALL PUTDEIO
CE54 2A 1A CC    404  LD HL,(FPOT2)
CE57 22 86 D1    405  LD (#FSZL),HL
CE5A 21 00 00    406  LD HL,0
CE5D 22 88 D1    407  LD (#FSZL+2),HL
CE60 3A 08 D1    408  LD A,(#FMAC)
CE63 E6 01       409  AND 1
CE65 3E 04       410  LD A,4
CE67 20 02       411  JR NZ,MACZOKU
CE69 3E 20       412     LD A,$20
CE6B             413 MACZOKU
CE6B 32 82 D0    414  LD (#ZOKU+2),A
CE6E 3E 02       415  LD A,2
CE70 32 80 D1    416  LD (#OD),A
CE73 CD 15 D0    417  CALL #SAVED
CE76             418 CLOSERET
CE76 DD E1       419  POP IX
CE78 E1          420  POP HL
CE79 D1          421  POP DE
CE7A C1          422  POP BC
CE7B F1          423  POP AF
CE7C C9          424  RET
CE7D             425
CE7D             426 ;
CE7D             427 CLRBF
CE7D 2A DF EC    428  LD HL,(#SBUFF)
CE80 ED 5B E3 EC 429  LD DE,(&BSIZ)
CE84 19          430  ADD HL,DE
CE85 ED 5B 5F CF 431  LD DE,(INBP2)
CE89 19          432  ADD HL,DE
CE8A EB          433  EX DE,HL
CE8B             434 CLRBFLP
CE8B AF          435     XOR A
CE8C CD 18 E0    436     CALL #LDDEA
CE8F 13          437     INC DE
CE90 2A E3 EC    438     LD HL,(&BSIZ)
CE93 B7          439     OR A
CE94 ED 52       440     SBC HL,DE
CE96 20 F3       441     JR NZ,CLRBFLP
CE98 C9          442  RET
CE99             443
CE99             444 ;
CE99             445 ;FREADS
CE99             446 ; (FILNO)
CE99             447 ;
CE99             448 READS
CE99 F5          449  PUSH AF
CE9A C5          450  PUSH BC
CE9B D5          451  PUSH DE
CE9C E5          452  PUSH HL
CE9D DD E5       453  PUSH IX
CE9F D5          454  PUSH DE
CEA0 06 FF       455  LD B,255
CEA2 21 00 EE    456  LD HL,P256
CEA5             457 READSLP
CEA5 11 64 CF    458     LD DE,EOFWK
CEA8 3E 01       459     LD A,1
CEAA 12          460     LD (DE),A
CEAB CD 06 CC    461     CALL FGETC
CEAE 1A          462     LD A,(DE)
CEAF 4F          463     LD C,A
CEB0 3A E7 EC    464     LD A,(&EOF)
CEB3 B9          465     CP C
CEB4 20 10       466     JR NZ,READSSK
CEB6 2A 18 CC    467        LD HL,(FPOT1)
CEB9 2B          468        DEC HL
CEBA 22 18 CC    469        LD (FPOT1),HL
CEBD 2A 5D CF    470        LD HL,(INBP1)
CEC0 23          471        INC HL
CEC1 22 5D CF    472        LD (INBP1),HL
CEC4 18 2D       473        JR READSRET
CEC6             474 READSSK
CEC6 79          475     LD A,C
CEC7 FE 0D       476     CP $0D
CEC9 20 1E       477     JR NZ,READSSK2
CECB 11 64 CF    478        LD DE,EOFWK
CECE 3E 01       479        LD A,1
CED0 12          480        LD (DE),A
CED1 CD 06 CC    481        CALL FGETC
CED4 1A          482        LD A,(DE)
CED5 FE 0A       483        CP $0A
CED7 28 1A       484        JR Z,READSRET
CED9 2A 18 CC    485           LD HL,(FPOT1)
CEDC 2B          486           DEC HL
CEDD 22 18 CC    487           LD (FPOT1),HL
CEE0 2A 5D CF    488           LD HL,(INBP1)
CEE3 23          489           INC HL
CEE4 22 5D CF    490           LD (INBP1),HL
CEE7 18 0A       491           JR READSRET
CEE9             492 READSSK2
CEE9 3A 8C E0    493     LD A,(#STOP)
CEEC B7          494     OR A
CEED 20 04       495     JR NZ,READSRET
CEEF 71          496     LD (HL),C
CEF0 23          497     INC HL
CEF1 10 B2       498     DJNZ READSLP
CEF3             499 READSRET
CEF3 78          500  LD A,B
CEF4 2F          501  CPL
CEF5 D1          502  POP DE
CEF6 12          503  LD (DE),A
CEF7 DD E1       504  POP IX
CEF9 E1          505  POP HL
CEFA D1          506  POP DE
CEFB C1          507  POP BC
CEFC F1          508  POP AF
CEFD C9          509  RET
CEFE             510
CEFE             511 ;
CEFE             512 ;FWRITS
CEFE             513 ; (DATA)
CEFE             514 ;
CEFE             515 WRITS
CEFE F5          516  PUSH AF
CEFF C5          517  PUSH BC
CF00 D5          518  PUSH DE
CF01 E5          519  PUSH HL
CF02 DD E5       520  PUSH IX
CF04 78          521  LD A,B
CF05 B7          522  OR A
CF06 28 0C       523  JR Z,WRITSRET
CF08             524 WRITSLP
CF08 CD 09 CC    525     CALL FPUTC
CF0B 3A 8C E0    526     LD A,(#STOP)
CF0E B7          527     OR A
CF0F 20 03       528     JR NZ,WRITSRET
CF11 13          529     INC DE
CF12 10 F4       530     DJNZ WRITSLP
CF14             531 WRITSRET
CF14 DD E1       532  POP IX
CF16 E1          533  POP HL
CF17 D1          534  POP DE
CF18 C1          535  POP BC
CF19 F1          536  POP AF
CF1A C9          537  RET
CF1B             538
CF1B             539 ;
CF1B             540 ;FREAD
CF1B             541 ; (LENGTH)
CF1B             542 ;
CF1B             543 READ
CF1B F5          544  PUSH AF
CF1C C5          545  PUSH BC
CF1D D5          546  PUSH DE
CF1E E5          547  PUSH HL
CF1F DD E5       548  PUSH IX
CF21 D5          549  PUSH DE
CF22 1A          550  LD A,(DE)
CF23 47          551  LD B,A
CF24 21 00 EE    552  LD HL,P256
CF27             553 READLP
CF27 11 64 CF    554     LD DE,EOFWK
CF2A 3E 01       555     LD A,1
CF2C 12          556     LD (DE),A
CF2D CD 06 CC    557     CALL FGETC
CF30 1A          558     LD A,(DE)
CF31 4F          559     LD C,A
CF32 3A E7 EC    560     LD A,(&EOF)
CF35 B9          561     CP C
CF36 20 10       562     JR NZ,READSK
CF38 2A 18 CC    563        LD HL,(FPOT1)
CF3B 2B          564        DEC HL
CF3C 22 18 CC    565        LD (FPOT1),HL
CF3F 2A 5D CF    566        LD HL,(INBP1)
CF42 23          567        INC HL
CF43 22 5D CF    568        LD (INBP1),HL
CF46 18 0A       569        JR READRET
CF48             570 READSK
CF48 3A 8C E0    571     LD A,(#STOP)
CF4B B7          572     OR A
CF4C 20 04       573     JR NZ,READRET
CF4E 71          574     LD (HL),C
CF4F 23          575     INC HL
CF50 10 D5       576     DJNZ READLP
CF52             577 READRET
CF52 D1          578  POP DE
CF53 1A          579  LD A,(DE)
CF54 90          580  SUB B
CF55 12          581  LD (DE),A
CF56 DD E1       582  POP IX
CF58 E1          583  POP HL
CF59 D1          584  POP DE
CF5A C1          585  POP BC
CF5B F1          586  POP AF
CF5C C9          587  RET
CF5D             588
CF5D             589 ;
CF5D             590 ;WORK DATA
CF5D             591 ;
CF5D 00 00       592 INBP1 DW 0
CF5F 00 00       593 INBP2 DW 0
CF61 00          594 IOFG1 DB 0
CF62 00 00       595 USRWK DW 0
CF64 00          596 EOFWK DB 0
CF65 00          597 FEOF  DB 0
CF66 00          598 EOR   DB 0
CF67             599
```

▶今月のディスクにはすごいものを入れてくれたものだ。４月号のディスクには何が入ってくるかいまから楽しみだ。３月18日までディスク５枚を用意しとくぜ。

古川　照道(17)栃木県

マシン語カクテル in Z80's Bar

第18回――乱数は世界を救うか――

シナリオ：西川善司
イラスト：山田純二

今月はひさびさに西川善司君がメインとなるマシン語カクテル。ちゃんと話が進行するのかどうかちょっと心配ですが，大丈夫でしょうか。まあ，ネタは乱数生成法ということですから，とりあえずでたらめに進めればいい？

季節外れの台風の日。
♪カラン，コローン（ドアベルの音）
……ゴーゴー（外の風の音）
……バタン（ドアの閉まる音）
西川善司（以下善）：おや，みなさんこんな台風の日にお揃いでどうしたんです？
源光（以下光）：とかなんとかいって自分だってやってきたくせに。
マスター(以下M)：やあ,善さん，いらっしゃい。今日は一体どうしたんですか。先月は1回も来なかったのに。
善：ぎくんちょ。
山田純二（以下純）：で，今日は……わかった。ゼンジソフトでまたくだらないものを作っていて行き詰まったんでしょう？
善：はふ，ちょっとため息。いえね，いまちょっとしたゲームを作っているんですがね。乱数の作り方がわかんないんですよ。
長老（以下老）：まったくお前というやつは，そんな基本も知らんでどうする！
純：「ランスウ」というと，あのアリスソフトの代表作？
光：それは「ランス」でしょ(しっかり知ってるやつ)。
善：あ,皆さんもやったんですかぁ。あれって結構面白いんですよね。私はシイルちゃんのファンなんですよ。へへ。
ようこ（以下Yo）：こらこら，そこ。変な話題で盛り上がらないように。

カミの味噌汁？

M：乱数って要するにでたらめな数の並びでしょ。作るのがそんなに難しいとは思えませんけどね。
老：一般にいう「乱数」というのは，実は「疑似乱数」のことなんじゃよ。完璧な乱数というものはまさに「神のみぞ知る」といったところじゃ。
善：「髪の味噌汁」？　うえーっまずそ。
Yo：あんたは黙ってなさい。
老：ま，コンピュータで作られる乱数の多くは「算術乱数」と呼ばれるものだ。つまり，ある「計算」によって求められた答えが「乱数」というわけじゃのう。
純：それでは，乱数に規則性や周期が現れてしまうのでは？
老：確かにな。じゃから，その「規則性」や「周期」がなるべく目立たない「疑似乱数」を作ろうと世界中がいまも研究をしとる。
光：私が聞いた話では円周率「π」や自然対数の底「e」などの無理数の各桁に出現する0～9の数字はほとんど同様の確率で，しかもでたらめの順番に出現していると聞いたことがありますね。
M：へぇー。それは確かに先ほどの「神のみぞ知る」乱数に近そうですね。
善：「紙の味噌汁」？　味噌汁はやっぱり豆腐とネギがいちばんだね。
Yo：善ちゃん，あなたここになにしに来たか覚えてる？
老：（無視して）それは，興味深い話題じゃがそれをもとにして乱数を作るにはちといまの計算機には重荷じゃろう。メモリや計算速度の点でな……。
純：「でたらめ」な数の並びがほしいんならなにかの「ノイズ」をソースにしたらどうでしょうか。
M：あ，確かに。「ノイズ（雑音）」レベルを数値に変換したら乱数になりそうですね。
善：「ホワイトノイズ」なんかは均等な周波数スペクトルを持ってるんだからフィルタやなんかで乱数の上限や下限を設定できるし，いいじゃないすかぁ。
老：確かに「ツェナダイオード（Zener diode)」などのノイズをソースに乱数を作ることはある。しかし，そういった乱数は「再現性」がないのはわかるじゃろう。
光：あ，なるほど。
Yo：「再現性」？
光：いいかい，ようこちゃん。ある数式実験で引数に乱数を使う場合を考える。そこで，ある乱数系列で実験結果が得られたとして，もう一度この実験を行うとき，「ノイズ」のような再現性のまったくない乱数では同じ結果はまず得られないのさ。ま，そうした点がこういう「ノイズ」型乱数の欠点だね。
善：どうでもいいけどさあ。早く具体的な乱数の発生法を教えてよ。はふはふ。ため息2連発出しちゃうよあたしゃ。
老：誰じゃお前は！
善：魅由と呼んでっ♥© 闇の血族

算術乱数

老：とても簡単で理解しやすいのが「平方採中法」(Middle sequence method)というやつじゃ。
M：名前は難しそうですけど……。
老：図1を見ながら話をしようか。まず，2進数aビットのある初期値を与える。この値を2乗し，その答えを（2×a）桁のレジスタに格納する。そして，その値の真ん中のaビットを求める乱数とし，以降，同じことを繰り返していく……というものじゃ。

純：なんか単純ですね。
善：でも，aがもし0になったら，以降0の繰り返しになっちゃうじゃん。
老：確かに。だから，得た乱数をそのまま次回の初期値とせず，Z80の場合ならリフレッシュレジスタなんかとXORなどをとるとよかろう。もっともそうするとさっき話した「乱数の再現性」というものがなくなるがな。
Yo：リフレッシュレジスタって？
光：RAMチップは一定期間ごとにメモリを読み出して書き込んでやらないと記憶された情報が消えてしまうんだ。まさにこの動作が「リフレッシュ」さ。もっともハードウェア的に行われるものだからユーザーはまったく気にしなくていいんだけどね。で，リフレッシュレジスタはこのリフレッシュ動作に使用するレジスタなんだよ。まあ，時間周期的にでたらめな値が入っていると考えてもいいんじゃないかな。
Yo：だったらそのまま乱数として使えそうじゃない？
光：でもZ80のリフレッシュレジスタは7ビットなんだよ。それにそのまま乱数として用いるにはいまいちバラツキがよくないんだな。
Yo：ふうん。
老：では，次に「線形合同法」(Linear congruential method)を紹介しようかのう。
善：……SIGH。これまた難しい名前だな。
老：何を言っとる。これは，そうじゃ，西川。おぬしがいつも愛用している「Oh！MZ乱数ルーチン」もこのタイプのものじゃぞ。(リスト6)
善：へぇ。
老：それで，この方式も比較的単純な原理のわりにはそこそこの結果を得られるので広く用いられている。数式1を見てもらおうか。初期値xをa倍しこれにcを加える。これだけじゃ。
Yo：小学生でもわかるわね。
光：ただし，出来のいい乱数をこの式から作るにはある条件のもとでパラメータa，cを定めなければならない。
善：で，その条件って？
光：話をLビット2進数の乱数（L≧2のとき）を求めることに場合をかぎれば，
1)　cと2^Lは互いに素である
2)　aは4の倍数に1を加えたものにする
とよい。
Yo：(恥ずかしそうに)……あの1番めの条件の「互いに素」ってどういう意味？
善：やだね，公約数が1以外はないっていう意味だよ。
Yo：ふーん？
老：Lビット2進数以外の範囲の乱数を求める話はここでは省略しておこう。数学の苦手な人もおるようじゃしな。
Yo：ふん！
善：いぬのふん，なんちて。ぽっくん。
一同：しらーっ……。
善：(空気を悟ってすかさず……)と，ところで長老あんまりいやみばっかいってると，いい死に方しませんぜ……。
老：わしはお前の墓参りをするまでは死なんぞ。
M：まあまあまあ。

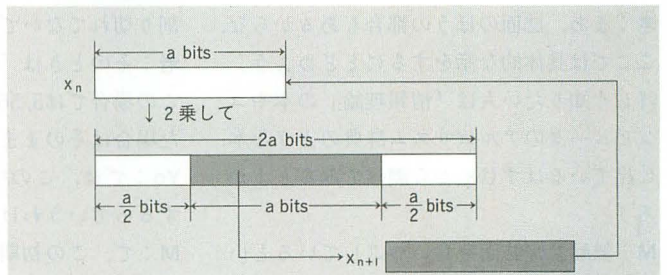



図1　平方採中法

数式1　線形合同法

$$x_{n+1}=ax_n+C$$

・Cと2^Lは互いに素
・aは4の倍数に1を加えたものにする

数式2　M系列法

パラメータp，q　　　（$1<q<p$）
$n=p/L$　　割り切れないときは$n=n+1$
$m=n\times L-p$

初期値演算
$$\begin{cases} x_{n-1}=((x_0 \text{ shr } m) \text{ XOR } x_{n-2}) \text{ OR } (x_{n-1} \text{ shl } m) \\ x_t=((x_{t-n+1} \text{ shr } m) \text{ OR } (x_{t-n} \text{ shl } m)) \text{ XOR } x_{t-1} \end{cases}$$

$$x_t=x_{t-p} \text{ XOR } x_{t-q}$$

乱数発生演算式

リスト1　平方採中法

```
0000                 1 ;     平方採中法(8BITS)
0000                 2 ;
E000                 3         ORG     0E000H
E000                 4
E000 FE 03           5         CP      3              ;CASE PARAMETERS ARE STR DATA
E002 28 08           6         JR      Z,INITIALIZE
E004 FE 02           7         CP      2              ;CASE PARAMETERS ARE INT DATA
E006 C0              8         RET     NZ             ;ERROR
E007 22 50 E0        9         LD      (HL_BUFF),HL   ;戻り値
E00A 18 08          10         JR      GET_RAND
E00C                11
E00C                12 INITIALIZE:
E00C 78             13         LD      A,B
E00D B7             14         OR      A
E00E C8             15         RET     Z              ;CASE NO PARAMETERS
E00F 1A             16         LD      A,(DE)         ;GET PARAMETERS FROM STR DATA
E010 32 52 E0       17         LD      (DATA),A
E013 C9             18         RET
E014                19
E014                20 GET_RAND:                      ;実際に乱数を得る
E014 2A 52 E0       21         LD      HL,(DATA)
E017 5D             22         LD      E,L
E018 54             23         LD      D,H
E019 CD 3E E0       24         CALL    KAKE           ;HL=HL^2
E01C                25
E01C ED 5F          26         LD      A,R
E01E AC             27         XOR     H
E01F AD             28         XOR     L
E020 08             29         EX      AF,AF'
E021                30
E021 CB 3C          31         SRL     H
E023 CB 1D          32         RR      L
E025 CB 3C          33         SRL     H
E027 CB 1D          34         RR      L
E029 CB 3C          35         SRL     H
E02B CB 1D          36         RR      L
E02D CB 3C          37         SRL     H
E02F CB 1D          38         RR      L              ;真ん中の8ビットをとる
E031                39
E031 7D             40         LD      A,L            ;A=RANDOM NUMBER
E032 2A 50 E0       41         LD      HL,(HL_BUFF)
E035 77             42         LD      (HL),A
E036 23             43         INC     HL
E037 36 00          44         LD      (HL),0
E039                45
E039 08             46         EX      AF,AF'
E03A 32 52 E0       47         LD      (DATA),A
E03D C9             48         RET
E03E                49
E03E                50 KAKE: ; HL*DE   ｺﾀｴ=HL
E03E 3E 10          51         LD      A,16
E040 44             52         LD      B,H
E041 4D             53         LD      C,L
E042 21 00 00       54         LD      HL,0
E045                55 ML1:
E045 29             56         ADD     HL,HL
E046 EB             57         EX      DE,HL
E047 29             58         ADD     HL,HL
E048 EB             59         EX      DE,HL
E049 30 01          60         JR      NC,ML2
E04B 09             61         ADD     HL,BC
E04C                62 ML2:
E04C 3D             63         DEC     A
E04D 20 F6          64         JR      NZ,ML1
E04F C9             65         RET
E050                66
E050 00 00          67 HL_BUFF:        DW      0
E052 00 00          68 DATA:           DW      0
```

M系列法

老：(気を取り直して) 3つめは「M系列法」(Maximum length recurring sequence method)というやつじゃ（数式2)。今度のはちと説明がややこしいぞ。さて，理論から説明するとすれば原始多項式やガロア体の話から始めないといけないんじゃが……（ようこちゃんのほうをちらっと見て)。

善：も，もう難しい話はたくさんだわ。

Yo：善ちゃん，人の声色で自分の意見をいわないでね。

老：まあ，誌面のほうの都合もあるからな，ここでは具体的な話をするにとどめよう。詳しく知りたい人は「情報理論」の本やコンピュータのアルゴリズム辞典のような本に出ているはずじゃから調べてみるとよかろう。

M：無駄話が誌面を食いつぶしているという話が……。

老：M系列法は2つのパラメータをまず用意する。仮にp, qとするが，この2つの値は正の数で，

$$1 < q < p$$

ならばなんでもよい。ただ，バラつきのよい乱数を出すのにいくつか経験的に知られているp，qの組み合わせがあるのだが，誰か知っておるかな？

一同：しーん。

老：$(p,q)=(521,32)$

$(p,q)=(250,103)$

$(p,q)=(89,38)$

などがよく使われるパラメータじゃ。ではいまから16ビット2進数の乱数を生成する場合を考えよう。このビット長をL＝16とするぞ。さて，(p,q)は……。

善：いちばん下の(89,38)でいこうよ。

老：よし。まずパラメータpをビット長Lで割る。

M：$p/L=89/16$

$=5.5625$

ですね。

老：この計算の答えの数だけ初期値を与える。

純：答えの数だけっていったって5.5625と，割り切れてないですけど。

老：そのときは「繰り上げ」をするのじゃ。この場合では5.5625→6とする。割り切れた場合はそのままでよいのじゃぞ。

Yo：では，この場合は6つ初期値を用意するっていうわけね。

M：で，この初期値はどうやって決めるんです？

老：適当でいいんじゃ。ただし，全部0とかはだめじゃぞ。

善：何ビットの値を用意すればいいの？

老：おお，忘れるとこじゃった。求めようとしている乱数のビット数と同じビット数の数じゃ。いまの場合だとL＝16でやっておるから16ビットじゃな。

善：て，いうことは$0\sim2^{16}-1$（0～65535）の範囲の数を6つ用意してこれを初期値にする，と。

老：そうじゃ。初期値を，

$$x_0, x_1, \cdots\cdots, x_5$$

として，初期値の個数をnとする。

（$x_0, x_1, \cdots\cdots, x_{n-1}$であることに注意)

このnをL倍したものからパラメータpを引く。

純：ええと，初期値の数はn＝6だから，

$n\times L-p=6\times16-89$

$=7$

老：よろしい。で，その答えをmとしよう。そして与えた初期値のうちの最後のx_{n-1}を次の要領で計算しなおす。

$x_{n-1}=((x_0 \text{ shr } m) \text{ XOR } x_{n-2}) \text{ OR } (x_{n-1} \text{ shl } m)$

いまの場合ではどうなるかな，西川君。

善：はふ。n＝6, m＝7でしょ，だから，

$x_5=((x_0 \text{ shr } 7) \text{ XOR } x_4) \text{ OR } (x_5 \text{ shl } 7)$

だね。

Yo：ええと。「XOR」や「OR」はわかるけど「shr」「shl」ってなあに？

純：それはね，ビットシフトさ。「shr」は右シフト，「shl」は左シフトを意味する。上の「x_0 shr 7」は「x_0」(のビットパターンを) を右へ「7」ビットシフトするってこと。

M：ああ，「C言語」の「<<」や「>>」と同じですね。

善：あのう。m＝0になっちゃったときはどうするんです？

初期値の数nを求めるときの，

$n=p/L$

の計算でnが割り切れちゃう場合は，

$m=n\times L-p$

は0になっちゃいますよ。

リスト2　線形合同法

```
0000              1 ;     線形合同法(8BITS)
0000              2 ;
E000              3        ORG     0E000H
E000              4
E000 FE 03        5        CP      3              ;CASE PARAMETERS ARE STR DATA
E002 28 08        6        JR      Z,INITIALIZE
E004 FE 02        7        CP      2              ;CASE PARAMETERS ARE INT DATA
E006 C0           8        RET     NZ             ;ERROR
E007 22 46 E0     9        LD      (HL_BUFF),HL   ;戻り値
E00A 18 0D       10        JR      GET_RAND
E00C             11
E00C             12 INITIALIZE:
E00C 78          13        LD      A,B
E00D B7          14        OR      A
E00E C8          15        RET     Z              ;CASE NO PARAMETERS
E00F 1A          16        LD      A,(DE)         ;GET PARAMETERS FROM STR DATA
E010 32 48 E0    17        LD      (PARAM_A),A
E013 13          18        INC     DE
E014 1A          19        LD      A,(DE)
E015 32 4A E0    20        LD      (PARAM_C),A
E018 C9          21        RET
E019             22
E019             23 GET_RAND:                     ;実際に乱数を得る
E019 2A 48 E0    24        LD      HL,(PARAM_A)
E01C ED 5B 4C    25        LD      DE,(LAST_RND)
E01F E0
E020 CD 34 E0    26        CALL    KAKE           ;HL=a*Xn-1
E023             27
E023 ED 5B 4A    28        LD      DE,(PARAM_C)
E026 E0
E027 19          29        ADD     HL,DE          ;HL=a*Xn-1+c
E028             30
E028 22 4C E0    31        LD      (LAST_RND),HL
E02B             32
E02B EB          33        EX      DE,HL          ;DE=RANDOM NUMBER
E02C 2A 46 E0    34        LD      HL,(HL_BUFF)
E02F 73          35        LD      (HL),E
E030 23          36        INC     HL
E031 36 00       37        LD      (HL),0         ;"0"を"D"にかえれば
E033 C9          38        RET                    ;16BITS乱数が得られる
E034             39
E034             40 KAKE: ; HL*DE  ｺﾀｴ=HL
E034 3E 10       41        LD      A,16
E036 44          42        LD      B,H
E037 4D          43        LD      C,L
E038 21 00 00    44        LD      HL,0
E03B             45 ML1:
E03B 29          46        ADD     HL,HL
E03C EB          47        EX      DE,HL
E03D 29          48        ADD     HL,HL
E03E EB          49        EX      DE,HL
E03F 30 01       50        JR      NC,ML2
E041 09          51        ADD     HL,BC
E042             52 ML2:
E042 3D          53        DEC     A
E043 20 F6       54        JR      NZ,ML1
E045 C9          55        RET
E046             56
E046 00 00       57 HL_BUFF:       DW      0
E048 00 00       58 PARAM_A:       DW      0
E04A 00 00       59 PARAM_C:       DW      0
E04C 21 85       60 LAST_RND:      DW      8521H          ;初期値(適当でいい)
```

老：実はそのときは，

$x_{n-1}=((x_0\ \mathrm{shr}\ m)\mathrm{XOR}\ x_{n-2})\mathrm{OR}\ (x_{n-1}\ \mathrm{shl}\ m)$

の計算はしなくていいのじゃ。と，いうのは，この計算は乱数周期を最大周期 2^{P-1} にするために，与えた初期値の足りないビット数を補足するための計算なんじゃ，といってもわけがわからんじゃろうが。

善：あんたみたいな大学教授をどっかの大学で見たことがあるよ。

Yo：あ，いるいる！　自分が頭のいいことを生徒に知らしめるがためにわけわかんないことというやつ。

M：……（ようこちゃんて結構口が悪い）。

老：いずれ，おぬしらにもわかる日がくるじゃろう。

純：(小声で）と，いやみにだめおしをする長老……。

老：なにかいったかな，純二君？　ふん。まあ，いいわい。次に最初に与えた初期値をもとにさらに $x_n \sim x_{p-1}$ の初期値を次の要領で計算する。

$x_t=((x_{t-n+1}\ \mathrm{shr}\ m)\mathrm{OR}(x_{t-n}\ \mathrm{shl}\ m))\mathrm{XOR}\ x_{t-1}\qquad(n\leqq t\leqq p-1)$

善：いまやっている例の場合だと，

$x_t=((x_{t-5}\ \mathrm{shr}\ 7)\mathrm{OR}(x_{t-6}\ \mathrm{shl}\ 7))\ \mathrm{XOR}\ x_{t-1}$

ですね。で，この漸化式で初期値を $x_6 \sim x_{89}$ まで求めればいいわけね。

M：すると，ずいぶん多くの初期値を必要とするんですね。じゃあ，先ほどのパラメータp,qの組み合わせで(521,32)というのがありましたけれど，あの場合だと初期値を $x_0 \sim x_{521}$ も用意しなくてはならないんですか。

老：そのとおり。しかし，その場合だと乱数周期は $2^{521}-1$，かなり周期の長い乱数が得られるな。

Yo：いまやってる例でも周期は $2^{89}-1$ ね。これでも充分長い周期よね。

善：さっきと同じくm＝0のときは上の計算はしなくていいの？

老：いや，m＝0のときは「shl」や「shr」を無視した，

$x_t=(x_{t-n+1}\ \mathrm{OR}\ x_{t-n})\mathrm{XOR}\ x_{t-1}\qquad(n\leqq t\leqq p-1)$

を計算すればよい。

純：で，初期値を用意して実際はどうやって乱数を得ればいいんです？

老：次の漸化式

$x_t=x_{t-p}\ \mathrm{XOR}\ x_{t-q}\qquad(p\leqq t)$

を順次計算し，その答えが乱数となる。

M：やっとqというパラメータが出てきたわけですね。

純：ええといまやっている例の場合だと

$x_t=x_{t-89}\ \mathrm{XOR}\ x_{t-38}\qquad(89\leqq t)$

ですね。

善：ほほう。この方式だと初めの計算量は多いけど，乱数を取り出すときは「XOR」1回でいいのか。この方法は高速だから，ゲームなんかにゃ向いてるな。おや，ちょっと待ってくれよ！　この例の場合，tが128以上の場合はどうすんのよ。

$x_t=x_{128-89}\ \mathrm{XOR}\ x_{128-38}$
$=x_{89}\ \mathrm{XOR}\ x_{90}$

となるけど初期値は90までしかないでしょうが！

老：はっはっは，愚かな。上の式はあくまで数学の漸化式で表現したものじゃ。実際のプログラムではt－p,t－qは単なる初期値の位置を示すポインタじゃから，得た x_t を順次t－pへ格納するようにして，t－p,t－qがpを超えるときはそれぞれを初期値のあるアドレスの先頭に戻してやればなんら問題はない。

善：……。

老：さて，ほかに乱数を求める方法には「逆関数法」，「棄却法」，「極座標法」などもあるが，ま，今日はこのへんでやめておくかな。興味のある人は……。

善：自分で調べよう，か。いい加減なもんだな。

そして，数時間後

善：できましたよ。リスト1が平方採中法，リスト2が線形合同法，リスト3と4がM系列法だよ。長老が説明したアルゴリズムのとおりプログラムしたから，特にプログラム自体の説明はしないよ。各リスト，注釈文と本文をよく照らし合わせてみればわかってもらえると思う。

老：どれどれ。なるほど。平方採中法にはリフレッシュレジスタを用いておるな。ということはリスト1で得られた乱数には再現性がないということじゃ。

善：リスト1と2,3は8ビット乱数が得られるんだ。リスト2のほうは一部を変更す

ると16ビット乱数を得られるようになるよ。リスト4はリスト3を16ビットに拡張したものだよ。

Yo：あら，じゃあリスト1の平方採中法の16ビット版はないのね。

善：だってさあ，もしそれをやるとしたら32ビットの乗算ルーチンを作らないといけないでしょ。そんなことするんならほかの方法を使ったほうがいいんじゃない？

老：で，どれも，BASICから使えるようなプログラムになっていると見たが？

善：ええ，そのとおり。BASICはX1のHuBASICかX1turboのTurboBASICのどちらかを使う。同じZ80のマシンへの移植はBASICのUSR文の仕様さえ合わせればかなり楽だと思うよ。

リスト1は，

A$=USR(CHR$(n))

のようにnに初期値を入れてコールすると，

A=USR(A)

で，Aに0～255の乱数が入ってくる。このときの変数Aは「整数型」でないとだめ。

リスト2は，

A$=USR(CHR$(a,c))

のように線形合同法のパラメータaとcを受け渡したあと，

A=USR(A)

でAに0～255の乱数が入ってくる。リスト後半に注釈があるけどそこを変更すれば16ビット，つまり0～65535の乱数が得られるようになるよ。

リスト3は，

A$=USR(CHR$(p,q))

でM系列法の2つのパラメータを受け渡したあと，

A=USR(A)

でAに0～255の乱数が入ってくる。さっ

きもいったけどリスト4はリスト3を16ビット拡張したもので，同じ使用方法で0～65535の乱数が得られる。

どのルーチンもあらかじめCLEAR &HE000を実行してからLOADMでオブジェクトを読み込んでおかないとダメだよ。DEFUSR＝&HE000も実行しておかないとダメ。

と，まあ，こんな感じ。BASICからでなくて機械語のプログラムから呼んだりするサブルーチンプログラムにするんならもうちょっと短くなるよね。

老：いやあずいぶんと長いセリフじゃったが，最後に，これらの乱数がどういったバラツキをするか視覚的に見てみようかの。

一同：え，なにそれ～っ。

乱数検定モドキ

老：乱数の検定にはいろいろあるがいちばん簡単でかつ興味深い結果をえられる方法を紹介しよう。とても簡単じゃからようこちゃんも大丈夫じゃよ。

Yo：ふん！

善：うまのふん。

老：2次元座標系の片方の軸に得た乱数X_J，もう片方の軸に1回前に得た乱数X_{J-1}をとり点を打っていくんじゃ。西川の作ってくれた乱数ルーチンをこの方法で見てみるとすればこんな感じかな（リスト5）。

このプログラムの使い方はまず，あらかじめ見てみたい乱数ルーチンを読み込んでおくことが前提じゃ。また，各乱数の発生法，ビット数を考慮してリスト5を変更したら「RUN」してみなさい。

：

Yo：あ，平方採中法やM系列法はそうでもないけど，線形合同法は斜めの線が出てきたわね。

老：これは得た乱数に規則性があるという証拠じゃ。リスト2中盤の，

```
    LD    (HL),E
    INC   HL
    LD    (HL),0
```

の部分を，

```
    LD    A,E
    XOR   D
    LD    (HL),A
    INC   HL
    LD    (HL),0
```

とするとかなりマシになるじゃろう。

Yo：あ，ほんとね。

善：注釈にもあるけど「LD (HL),0」の部分を「LD (HL),D」にすると16ビット乱数が得られるようになるよ。

老：16ビットの場合だと変な小細工をしなくてもかなりバラツキのよいものになっとるようじゃな。

純：……あ，M系列法もパラメータの選び方によっては変な「V」のような軌跡ができてしまいますね。

M：平方採中法は「リフレッシュレジスタ」を使用している分，結構いいバラツキのようです。

老：この方法でBASIC内蔵の乱数コマンドやリスト6の「Oh!MZ乱数ルーチン」なども検定してみるよい。リスト6は線形合同法なのでリスト2同様斜め模様ができてくるはずじゃ。リスト6は16ビットのものになっとるがリスト2同様の変更で8ビットのものにもなるぞ。

Yo：ところで光君は？

善：あ，ここで眠っているよ。

純：どおりで声がしなくなったわけだ。

光：ぶつぶつぶつ。

M：おや，なんか寝言いってますよ。

光：ようこちゃん……，むにゃ。

Yo：あら，やだ。私の名前だわ（ぽっ）。

光：……メアリー，くみこちゃん，ゆかりちゃん，しずかちゃん，さゆりちゃん，まりちゃん……。

純：なんか，かたっぱしから女性の名前をつぶやいてるぞ。まったく，この男は。

Yo：ん，もう。

善：JESUS……，なんちゃって。オチが読まれたかな，へへへへ。バビンチョ。

つづく

リスト3　M系列法

```
0000                 1 ;      M系列乱数(8BITS)
0000                 2 ;
E000                 3        ORG     0E000H
E000                 4
E000 FE 03           5        CP      3
E002 28 09           6        JR      Z,INITIALIZE
E004 FE 02           7        CP      2
E006 C0              8        RET     NZ               ;ERROR
E007 22 0D E1        9        LD      (HL_BUFF),HL     ;戻り値
E00A C3 B5 E0       10        JP      GET_RAND
E00D                11
E00D                12 INITIALIZE:
E00D 78             13        LD      A,B
E00E B7             14        OR      A
E00F 28 09          15        JR      Z,TRANS          ;CASE NO PARAMETERS
E011 1A             16        LD      A,(DE)           ;GET PARAMETERS FROM STR DATA
E012 32 05 E1       17        LD      (P),A
E015 13             18        INC     DE
E016 1A             19        LD      A,(DE)
E017 32 06 E1       20        LD      (Q),A
E01A                21 TRANS:
E01A 21 0F E1       22        LD      HL,SHOKICHI
E01D 11 35 E1       23        LD      DE,ESTBN
E020 01 28 00       24        LD      BC,40
E023 ED B0          25        LDIR
E025                26
E025 3A 05 E1       27        LD      A,(P)
E028 47             28        LD      B,A              ;B=P
E029 CB 3F          29        SRL     A                ;/2
E02B CB 3F          30        SRL     A                ;/4
E02D CB 3F          31        SRL     A                ;/8
E02F 30 01          32        JR      NC,MRAND0        ;割りきれた
E031 3C             33        INC     A                ;割りきれなかった
E032                34 MRAND0:
E032 32 04 E1       35        LD      (N),A            ;SAVE N
E035 87             36        ADD     A,A
E036 87             37        ADD     A,A
E037 87             38        ADD     A,A              ;*8
E038 90             39        SUB     B                ;N*L-P
E039 32 03 E1       40        LD      (M),A            ;SAVE M
E03C                41
E03C 3A 04 E1       42        LD      A,(N)
E03F 3D             43        DEC     A                ;A=N-1
E040 5F             44        LD      E,A
E041 16 00          45        LD      D,0
E043 DD 21 35       46        LD      IX,ESTBN
E046 E1
E047 DD 19          47        ADD     IX,DE
E049 DD 6E 00       48        LD      L,(IX+0)         ;HL=Xn-1
E04C DD 4E FF       49        LD      C,(IX-1)         ;BC=Xn-2
E04F                50
E04F 3A 03 E1       51        LD      A,(M)
E052 B7             52        OR      A
E053 28 13          53        JR      Z,INIT2          ;IF CASE M=0 THEN JUMP TO …
E055                54                                 ;Xn-1 shl m
E055 CD F8 E0       55        CALL    SHL_L
E058                56
E058 3A 35 E1       57        LD      A,(ESTBN)
E05B 5F             58        LD      E,A
E05C                59                                 ;X0 shr m
E05C CD ED E0       60        CALL    SHR_E
E05F                61                                 ;X0 XOR Xn-2 --> E
E05F 7B             62        LD      A,E
E060 A9             63        XOR     C
E061 5F             64        LD      E,A
E062                65                                 ;E OR Xn-1 --> L
E062 7D             66        LD      A,L
E063 B3             67        OR      E
E064 6F             68        LD      L,A
E065                69
E065 DD 75 00       70        LD      (IX+0),L         ;SAVE Xn-1
E068                71 INIT2:
E068 DD 23          72        INC     IX
E06A FD 21 35       73        LD      IY,ESTBN
E06D E1
E06E 3A 04 E1       74        LD      A,(N)
E071 47             75        LD      B,A
E072 3A 05 E1       76        LD      A,(P)
E075 90             77        SUB     B
E076 47             78        LD      B,A              ;B=計算回数
E077                79 INIT2_LOOP:                     ;残こりの初期値を計算する
E077 FD 6E 00       80        LD      L,(IY+0)
E07A CD F8 E0       81        CALL    SHL_L
E07D FD 5E 01       82        LD      E,(IY+1)
E080 CD ED E0       83        CALL    SHR_E
E083 7D             84        LD      A,L
E084 B3             85        OR      E
E085 DD AE FF       86        XOR     (IX-1)
E088 DD 77 00       87        LD      (IX+0),A
E08B DD 23          88        INC     IX
E08D FD 23          89        INC     IY
```

```
E08F 10 E6       90          DJNZ    INIT2_LOOP
E091             91                                       ;各ポインタの計算
E091 11 35 E1    92          LD      DE,ESTBN
E094 21 00 00    93          LD      HL,0
E097 ED 53 07    94          LD      (XP),DE              ;XP=0
E09A E1
E09B 3A 06 E1    95          LD      A,(Q)
E09E 47          96          LD      B,A                  ;B=Q
E09F 3A 05 E1    97          LD      A,(P)                ;A=P
E0A2 90          98          SUB     B                    ;A=P-Q
E0A3 6F          99          LD      L,A
E0A4 26 00      100          LD      H,0
E0A6 19         101          ADD     HL,DE
E0A7 22 09 E1   102          LD      (XQ),HL
E0AA            103                                       ;ポインタMAXの計算
E0AA 3A 05 E1   104          LD      A,(P)
E0AD 6F         105          LD      L,A
E0AE 26 00      106          LD      H,0
E0B0 19         107          ADD     HL,DE
E0B1 22 0B E1   108          LD      (XMAX),HL
E0B4 C9         109          RET
E0B5            110
E0B5            111 GET_RAND:                             ;実際に乱数を得る
E0B5 2A 07 E1   112          LD      HL,(XP)
E0B8 ED 5B 09   113          LD      DE,(XQ)
E0BB E1
E0BC            114
E0BC 1A         115          LD      A,(DE)               ;乱数の計算
E0BD AE         116          XOR     (HL)                 ;(これだけ!!)
E0BE            117
E0BE 77         118          LD      (HL),A               ;得た乱数を
E0BF            119                                       ;初期値TABLEにしまう
E0BF 08         120          EX      AF,AF'
E0C0 23         121          INC     HL
E0C1 13         122          INC     DE
E0C2 ED 4B 0B   123          LD      BC,(XMAX)
E0C5 E1
E0C6            124
E0C6 7D         125          LD      A,L
E0C7 B9         126          CP      C
E0C8 20 07      127          JR      NZ,SAVE_HL
E0CA 7C         128          LD      A,H
E0CB B8         129          CP      B
E0CC 20 03      130          JR      NZ,SAVE_HL
E0CE 21 35 E1   131          LD      HL,ESTBN
E0D1            132 SAVE_HL:
E0D1 22 07 E1   133          LD      (XP),HL
E0D4            134
E0D4 7B         135          LD      A,E
E0D5 B9         136          CP      C
E0D6 20 07      137          JR      NZ,SAVE_DE
E0D8 7A         138          LD      A,D
E0D9 B8         139          CP      B
E0DA 20 03      140          JR      NZ,SAVE_DE
E0DC 11 35 E1   141          LD      DE,ESTBN
E0DF            142 SAVE_DE:
E0DF ED 53 09   143          LD      (XQ),DE
E0E2 E1
E0E3 EB         144          EX      DE,HL                ;DE=RANDOM NUMBER
E0E4 2A 0D E1   145          LD      HL,(HL_BUFF)
E0E7 08         146          EX      AF,AF'
E0E8 77         147          LD      (HL),A
E0E9 23         148          INC     HL
E0EA 36 00      149          LD      (HL),0
E0EC C9         150          RET
E0ED            151
E0ED            152 SHR_E:
E0ED 3A 03 E1   153          LD      A,(M)
E0F0 B7         154          OR      A
E0F1 C8         155          RET     Z
E0F2            156 SHR_LP:
E0F2 CB 3B      157          SRL     E
E0F4 3D         158          DEC     A
E0F5 20 FB      159          JR      NZ,SHR_LP
E0F7 C9         160          RET
E0F8            161 SHL_L:
E0F8 3A 03 E1   162          LD      A,(M)
E0FB B7         163          OR      A
E0FC C8         164          RET     Z
E0FD            165 SHL_LP:
E0FD CB 25      166          SLA     L
E0FF 3D         167          DEC     A
E100 20 FB      168          JR      NZ,SHL_LP
E102 C9         169          RET
E103            170
E103 00         171 M:       DS      1
E104 00         172 N:       DS      1
E105 59         173 P:       DB      89
E106 26         174 Q:       DB      38
E107 00 00      175 XP:      DW      0
E109 00 00      176 XQ:      DW      0
E10B 00 00      177 XMAX:    DW      0
E10D 00 00      178 HL_BUFF:         DW      0
E10F            179 SHOKICHI:        ;初期値TABLE     適当で構いません(40個くらい)
E10F 0C 22 05   180          DB      12,34,5,67,89,10,11,12,131,4
E112 43 59 0A
E115 0B 0C 83
E118 04
E119 97 06 11   181          DB      151,6,17,18,19,20,212,22,3,24
E11C 12 13 14
E11F D4 16 03
E122 18
E123 7B 2D 43   182          DB      123,45,67,89,101,112,13,14,15,16
E126 59 65 70
E129 0D 0E 0F
E12C 10
E12D AB 51 5C   183          DB      171,81,92,21,222,32,45,99
E130 15 DE 20
E133 2D 63
E135            184 ESTBN:
E135 00 00 00   185          DS      256
```

リスト4　M系列法16ビット版

```
0000              1 ;    M系列乱数(16BITS)
0000              2 ;
E000              3          ORG     0E000H
E000              4
E000 FE 03        5          CP      3
E002 28 09        6          JR      Z,INITIALIZE
E004 FE 02        7          CP      2
E006 C0           8          RET     NZ                   ;ERROR
E007 22 66 E1     9          LD      (HL_BUFF),HL         ;戻り値
E00A C3 DF E0    10          JP      GET_RAND
E00D             11
E00D             12 INITIALIZE:
E00D 78          13          LD      A,B
E00E B7          14          OR      A
E00F 28 09       15          JR      Z,TRANS              ;CASE NO PARAMETERS
E011 1A          16          LD      A,(DE)               ;GET PARAMETERS FROM STR DATA
E012 32 5E E1    17          LD      (P),A
E015 13          18          INC     DE
E016 1A          19          LD      A,(DE)
E017 32 5F E1    20          LD      (Q),A
E01A             21 TRANS:
E01A 21 68 E1    22          LD      HL,SHOKICHI
E01D 11 B8 E1    23          LD      DE,ESTBN
E020 01 28 00    24          LD      BC,40
E023 ED B0       25          LDIR
E025             26
E025 3A 5E E1    27          LD      A,(P)
E028 47          28          LD      B,A                  ;B=P
E029 CB 3F       29          SRL     A                    ;/2
E02B CB 3F       30          SRL     A                    ;/4
E02D CB 3F       31          SRL     A                    ;/8
E02F CB 3F       32          SRL     A                    ;/16
E031 D2 35 E0    33          JP      NC,MRAND0            ;割りきれた
E034 3C          34          INC     A                    ;割りきれなかった
E035             35 MRAND0:
E035 32 5D E1    36          LD      (N),A                ;SAVE N
E038 87          37          ADD     A,A
E039 87          38          ADD     A,A
E03A 87          39          ADD     A,A
E03B 87          40          ADD     A,A                  ;*16
E03C 90          41          SUB     B                    ;N*L-P
E03D 32 5C E1    42          LD      (M),A                ;SAVE M
E040             43
E040 3A 5D E1    44          LD      A,(N)
E043 3D          45          DEC     A                    ;A=N-1
E044 6F          46          LD      L,A
E045 26 00       47          LD      H,0
E047 29          48          ADD     HL,HL                ;HL=A*2
E048 EB          49          EX      DE,HL
E049 DD 21 B8    50          LD      IX,ESTBN
E04C E1
E04D DD 19       51          ADD     IX,DE
E04F DD 6E 00    52          LD      L,(IX+0)
E052 DD 66 01    53          LD      H,(IX+1)             ;HL=Xn-1
E055 DD 46 FF    54          LD      B,(IX-1)
E058 DD 4E FE    55          LD      C,(IX-2)             ;BC=Xn-2
E05B             56
E05B 3A 5C E1    57          LD      A,(M)
E05E B7          58          OR      A
E05F 28 19       59          JR      Z,INIT2              ;IF CASE M=0 THEN JUMP TO …
E061             60                                       ;Xn-1 shl m
E061 CD 4F E1    61          CALL    SHL_HL
E064             62
E064 ED 5B B8    63          LD      DE,(ESTBN)
E067 E1
E068             64                                       ;X0 shr m
E068 CD 42 E1    65          CALL    SHR_DE
E06B             66                                       ;X0 XOR Xn-2 --> DE
E06B 7B          67          LD      A,E
E06C A9          68          XOR     C
E06D 5F          69          LD      E,A
E06E 7A          70          LD      A,D
E06F A8          71          XOR     B
E070 57          72          LD      D,A
E071             73                                       ;DE OR Xn-1 --> HL
E071 CD 3B E1    74          CALL    OR_HL_DE
E074             75
E074 DD 75 00    76          LD      (IX+0),L             ;SAVE Xn-1
E077 DD 74 01    77          LD      (IX+1),H
E07A             78 INIT2:
E07A DD 23       79          INC     IX
E07C DD 23       80          INC     IX
E07E FD 21 B8    81          LD      IY,ESTBN
E081 E1
E082 3A 5D E1    82          LD      A,(N)
E085 47          83          LD      B,A
E086 3A 5E E1    84          LD      A,(P)
E089 90          85          SUB     B
E08A 47          86          LD      B,A
E08B             87 INIT2_LOOP:                           ;残こりの初期値を計算する
E08B FD 6E 00    88          LD      L,(IY+0)
E08E FD 66 01    89          LD      H,(IY+1)
E091 CD 4F E1    90          CALL    SHL_HL
E094 FD 5E 02    91          LD      E,(IY+2)
E097 FD 56 03    92          LD      D,(IY+3)
E09A CD 42 E1    93          CALL    SHR_DE
E09D CD 3B E1    94          CALL    OR_HL_DE
E0A0 DD 56 FF    95          LD      D,(IX-1)
E0A3 DD 5E FE    96          LD      E,(IX-2)
E0A6 CD 34 E1    97          CALL    XOR_HL_DE
E0A9             98
E0A9 DD 75 00    99          LD      (IX+0),L
E0AC DD 74 01   100          LD      (IX+1),H
E0AF DD 23      101          INC     IX
E0B1 DD 23      102          INC     IX
E0B3 FD 23      103          INC     IY
E0B5 FD 23      104          INC     IY
E0B7 10 D2      105          DJNZ    INIT2_LOOP
E0B9            106                                       ;各ポインタの計算
E0B9 11 B8 E1   107          LD      DE,ESTBN
E0BC 21 00 00   108          LD      HL,0
E0BF ED 53 60   109          LD      (XP),DE              ;XP=0
E0C2 E1
E0C3 3A 5F E1   110          LD      A,(Q)
E0C6 47         111          LD      B,A                  ;B=Q
E0C7 3A 5E E1   112          LD      A,(P)                ;A=P
E0CA 90         113          SUB     B                    ;A=P-Q
E0CB 6F         114          LD      L,A
E0CC 26 00      115          LD      H,0
E0CE 29         116          ADD     HL,HL                ;HL=A*2
E0CF 19         117          ADD     HL,DE
E0D0 22 62 E1   118          LD      (XQ),HL
E0D3            119                                       ;ポインタMAXの計算
E0D3 3A 5E E1   120          LD      A,(P)
E0D6 6F         121          LD      L,A
E0D7 26 00      122          LD      H,0
E0D9 29         123          ADD     HL,HL                ;HL=A*2
E0DA 19         124          ADD     HL,DE
E0DB 22 64 E1   125          LD      (XMAX),HL
E0DE C9         126          RET
E0DF            127
E0DF            128 GET_RAND:                             ;実際に乱数を得る
E0DF DD 2A 60   129          LD      IX,(XP)
E0E2 E1
```

```
E0E3 FD 2A 62   130         LD      IY,(XQ)
E0E6 E1
E0E7            131
E0E7 DD 5E 00   132         LD      E,(IX+0)
E0EA DD 56 01   133         LD      D,(IX+1)
E0ED FD 6E 00   134         LD      L,(IY+0)
E0F0 FD 66 01   135         LD      H,(IY+1)
E0F3            136
E0F3 CD 34 E1   137         CALL    XOR_HL_DE
E0F6            138
E0F6 DD 75 00   139         LD      (IX+0),L        ;得た乱数を
E0F9 DD 74 01   140         LD      (IX+1),H        ;初期値TABLEにしまう
E0FC DD 23      141         INC     IX
E0FE DD 23      142         INC     IX
E100 FD 23      143         INC     IY
E102 FD 23      144         INC     IY
E104 ED 5B 64   145         LD      DE,(XMAX)
E107 E1
E108            146
E108 DD         147         DB      0DDH            ;ポインタ1処理
E109 7D         148         LD      A,L
E10A BB         149         CP      E
E10B 20 09      150         JR      NZ,SAVE_IX
E10D DD         151         DB      0DDH
E10E 7C         152         LD      A,H
E10F BA         153         CP      D
E110 20 04      154         JR      NZ,SAVE_IX
E112 DD 21 B8   155         LD      IX,ESTBN
E115 E1
E116            156 SAVE_IX:
E116 DD 22 60   157         LD      (XP),IX
E119 E1
E11A            158
E11A FD         159         DB      0FDH            ;ポインタ2処理
E11B 7D         160         LD      A,L
E11C BB         161         CP      E
E11D 20 09      162         JR      NZ,SAVE_IY
E11F FD         163         DB      0FDH
E120 7C         164         LD      A,H
E121 BA         165         CP      D
E122 20 04      166         JR      NZ,SAVE_IY
E124 FD 21 B8   167         LD      IY,ESTBN
E127 E1
E128            168 SAVE_IY:
E128 FD 22 62   169         LD      (XQ),IY
E12B E1
E12C EB         170         EX      DE,HL           ;DE=RANDOM NUMBER
E12D 2A 66 E1   171         LD      HL,(HL_BUFF)
E130 73         172         LD      (HL),E
E131 23         173         INC     HL
E132 72         174         LD      (HL),D
E133 C9         175         RET
E134            176
E134            177 XOR_HL_DE:
E134 7D         178         LD      A,L
E135 AB         179         XOR     E
E136 6F         180         LD      L,A
E137 7C         181         LD      A,H
E138 AA         182         XOR     D
E139 67         183         LD      H,A
E13A C9         184         RET
E13B            185 OR_HL_DE:
E13B 7D         186         LD      A,L
E13C B3         187         OR      E
E13D 6F         188         LD      L,A
E13E 7C         189         LD      A,H
E13F B2         190         OR      D
E140 67         191         LD      H,A
E141 C9         192         RET
E142            193 SHR_DE:
E142 3A 5C E1   194         LD      A,(M)
E145 B7         195         OR      A
E146 C8         196         RET     Z
E147            197 SHR_LP:
E147 CB 3A      198         SRL     D
E149 CB 1B      199         RR      E
E14B 3D         200         DEC     A
E14C 20 F9      201         JR      NZ,SHR_LP
E14E C9         202         RET
E14F            203 SHL_HL
E14F 3A 5C E1   204         LD      A,(M)
E152 B7         205         OR      A
E153 C8         206         RET     Z
E154            207 SHL_LP:
E154 CB 25      208         SLA     L
E156 CB 14      209         RL      H
E158 3D         210         DEC     A
E159 20 F9      211         JR      NZ,SHL_LP
E15B C9         212         RET
E15C            213
E15C 00         214 M:      DS      1
E15D 00         215 N:      DS      1
E15E 59         216 P:      DB      89
E15F 26         217 Q:      DB      38
E160 00 00      218 XP:     DW      0
E162 00 00      219 XQ:     DW      0
E164 00 00      220 XMAX:   DW      0
E166 00 00      221 HL_BUFF:        DW      0
E168            222 SHOKICHI:       ;初期値TABLE     適当で構いません(40個くらい)
E168 D2 04 37   223         DW      1234,567,8910,1112,1314,1516
E16B 02 CE 22
E16E 58 04 22
E171 05 EC 05
E174 B6 06 80   224         DW      1718,1920,2122,2324,12115,1
E177 07 4A 08
E17A 14 09 53
E17D 2F 01 00
E180 1D 02 35   225         DW      541,565,123,1567,1865,1861
E183 02 7B 00
E186 1F 06 49
E189 07 45 07
E18C 7B 00 9A   226         DW      123,154,189,10,112,5855,12345
E18F 00 BD 00
E192 0A 00 70
E195 00 DF 16
E198 39 30
E19A 7A 00 DF   227         DW      122,223,520,9999,229,19,542
E19D 00 08 02
E1A0 0F 27 E5
E1A3 00 13 00
E1A6 1E 02
E1A8 41 01 D2   228         DW      321,210,12,11,4,888,9876,543
E1AB 00 0C 00
E1AE 0B 00 04
E1B1 00 78 03
E1B4 94 26 1F
E1B7 02
E1B8            229 ESTBN:
E1B8 00 00 00   230         DS      256*2
E1BB 00 00 00
E1BE 00 00 00
E1C1 00 00 00
E1C4 00 00 00
E1C7 00 00 00
E1CA 00 00 00
E1CD 00 00 00
E1D0 00 00 00
```

リスト5 乱数検定プログラム

```
10 DEFINT A-Z:CLS4:INIT:CLEAR &HE000:DEFUSR=&HE000
20 'Example parameters
30  A$=USR(CHR$(99))              '平方採中法 8BITS
40 'A$=USR(CHR$(49,11))           '線形合同法 8BITS
50 'A$=USR(CHR$(39,11))           'M系列      8BITS
60 'A$=USR(CHR$(29,13))           'M系列     16BITS
70 'A$=USR(CHR$(&H12,&HE1))       'Oh!mz乱数 16BITS
80 'WINDOW(0,0)-(639,199),(-32768!,32768!)-(32768!,-32768!)      '16BITS乱数の時
90  WINDOW(0,0)-(639,199),(0,0)-(255,255)                        '8BITS乱数の時
100 B=USR(B)
110 '***** LOOP *****
120 A=USR(A)
130 PSET(A,B,1):B=A
140 GOTO 120
```

リスト6 オマケのサンプル

```
0000             1 ;    Oh!mz(16BITS) (その昔マシン語体操で出てきたもの)
0000             2 ;
E000             3          ORG     0E000H
E000             4
E000 FE 03       5          CP      3               ;CASE PARAMETERS ARE STR DATA
E002 28 08       6          JR      Z,INITIALIZE
E004 FE 02       7          CP      2               ;CASE PARAMETERS ARE INT DATA
E006 C0          8          RET     NZ              ;ERROR
E007 22 3F E0    9          LD      (HL_BUFF),HL    ;戻り値
E00A 18 0C      10          JR      GET_RAND
E00C            11
E00C            12 INITIALIZE:
E00C 78         13          LD      A,B
E00D B7         14          OR      A
E00E C8         15          RET     Z               ;CASE NO PARAMETERS
E00F 1A         16          LD      A,(DE)          ;GET PARAMETERS FROM STR DATA
E010 6F         17          LD      L,A
E011 13         18          INC     DE
E012 1A         19          LD      A,(DE)
E013 67         20          LD      H,A
E014 22 41 E0   21          LD      (OLDRND),HL
E017 C9         22          RET
E018            23
E018            24 GET_RAND:                        ;実際に乱数を得る
E018 ED 5B 41   25          LD      DE,(OLDRND)
E01B E0
E01C ED 4B 43   26          LD      BC,(STEP)
E01F E0
E020 CD 2E E0   27          CALL    MULTI
E023 22 41 E0   28          LD      (OLDRND),HL
E026            29
E026 EB         30          EX      DE,HL
E027 2A 3F E0   31          LD      HL,(HL_BUFF)
E02A 73         32          LD      (HL),E
E02B 23         33          INC     HL
E02C 72         34          LD      (HL),D
E02D C9         35          RET
E02E            36
E02E            37 MULTI:
E02E 21 00 00   38          LD      HL,0
E031 3E 10      39          LD      A,10H
E033            40 MLOOP:
E033 29         41          ADD     HL,HL
E034 CB 23      42          SLA     E
E036 CB 12      43          RL      D
E038 30 01      44          JR      NC,SKIP
E03A 09         45          ADD     HL,BC
E03B            46 SKIP:
E03B 3D         47          DEC     A
E03C 20 F5      48          JR      NZ,MLOOP
E03E C9         49          RET
E03F            50
E03F 00 00      51 HL_BUFF:         DW      0
E041 23 E1      52 OLDRND:  DW      0E123H
E043 83 03      53 STEP:    DW      899
```

THE SENTINEL

●実数型コンパイラへの想い

S-OS上のコンパイラの歩みはFuzzyBASICコンパイラから始まりました（その前にmagiFORTHという変わり種はありましたが）。FuzzyBASICで作成したプログラムをマシン語に変換することが可能となったのです。このあと1988年1月号ではFuzzyBASICで書かれたFuzzyBASICコンパイラさえ登場しています。

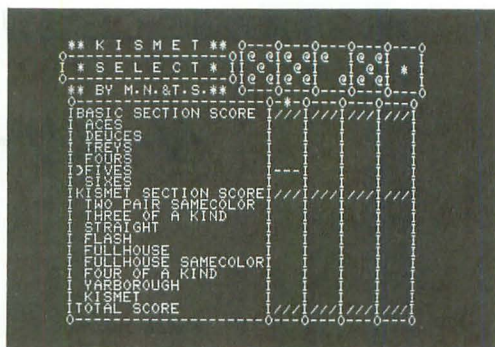
S-OS上のコンパイラの主流は構造化コンパイラ言語SLANGに引き継がれました。Cに似た文法を持つSLANGはいまではすっかりS-OS上の標準コンパイラとしての地位を確立した感があります。寄せられる投稿の多くがSLANGで書かれているということからも，人気の高さがうかがえます。

SLANGは整数型コンパイラです。実数演算パッケージSOROBAN用のライブラリが追加され，実数を扱えるようになったとはいっても，「1.0+2.0」を計算するのに，まず1と2を浮動小数点数に変換する関数を呼び出し，次に2つの数を足す関数を呼び出さなければならず，扱って扱えなくはないというレベルです。実数を直接扱えるコンパイラの登場が望まれるところです。

第104部
ダイスゲームKISMET

●ダイスゲームKISMET

一風変わったゲームが届きました。サイコロを5つ使ったダイスゲームKISMETです。出たサイコロの目によって役が決まり，役に応じた得点が入ります。役は全部で15種類で，BASICセクションとKISMETセクションの2つに分けられます。BASICセクションはサイコロの目を集計するもので，1の目だけを集計するACESから6の目だけを集計するSIXESまでの6種類。KISMETセクションはポーカーの役によく似た役が9種類用意されています。ゲームはサイコロを15回振り，できた役の合計得点で争います。

このゲームの面白いところは，サイコロを振って出た目を必ずなんらかの役にしなければならないこと。そして，同じ役はゲーム中二度と使えないことです。1・2・3・4・5と目が出たのに，すでにストレートの役を使ってしまっているときなどは悲劇です。ACESでは（1がひとつしかないので）1点しか入りませんし，SIXESにして6・6・5・4・3なんて役が次に出た日には目も当てられません。どの役を割り振るか，この選択が運命の別れ目となります。どうぞ悩み抜いてください。

●S-OSの系譜（18）

S-OSは全機種共通のBIOS（Basic Input Output System）＋マシン語モニタという体裁でスタートしました。X68000でいうならばBIOSはIOCS，マシン語モニタはデバッガといったところでしょうか。ただ，マシン語モニタにはプログラムをデバッグする機能はなく，マシン語プログラムの入力，ロード/セーブなど限られた機能が与えられていたにすぎません。

それを最もよく表しているのがプログラムの実行方法です。マシン語プログラムは，まずプログラムをメモリに読み込み，それからスタート番地にジャンプするという方法で実行されました。S-OSのモニタはマシン語プログラムをロードするアドレスを自分で指定できるようになっています。Z80のマシン語プログラムながらリロケータブルに作られたMACINTO-Sは，この機能を利用してユーザーが自分の好きなアドレスで実行できるようになっていました。

“SWORD”になってディスクを扱えるようになっても，この部分は変わりませんでした。シェルではなくモニタなんだという信念が開発者にあったからです。ただ，これでは少々使いづらいのではないか，という提言もあり，1986年10月号でRUNコマンドが追加されたのはこのコーナーですでに紹介したとおりです。

そして1987年5月号では，FuzzyBASICの作者である瀧山氏によりさらなる拡張が施されました。RUNコマンドはプログラムのロード・ジャンプをひとつの命令で行うものでしたが，さらにパラメータをプログラムに渡すことができるように改良されたのです。RUNコマンドはコマンドライン先頭にRを入力していましたが，これはスペースに変更されました。この改造によってプログラムの実行は，プログラム名だけを入力する，CP/MやHumanのような方法になったわけです。同時にRAMディスクもサポートされ，これまでテープでイライラしながら起動を待っていたプログラムの高速起動が実現されました。

全機種共通S-OS“SWORD”要

ダイスゲームKISMET

サイコロを使ったポーカーという風合いのゲーム，KISMETです。ちなみにKISMETの意味は「幸運」。でも高得点を狙うにはかなりテクニックも必要とされるみたいですね。このゲームを実行するにはSLANGが必要です。

Sakaki Takuya
榊　卓也

ゲームの概要

SLANGコンパイラを使ったダイスゲームです。5個のサイコロを振って15の役を作り，総得点を競うゲームです。あまり知られていないゲームですので，最初に概要を説明しておきましょう。

サイコロを使用します。普通のサイコロでもできなくはないのですが，このゲームで使うサイコロは目が3色で塗り分けられています。1，6が青，2，5が赤，3，4が緑です。ゲームは1～4人で行います。まず，名前を登録し，最初の人からこのようなサイコロを5個同時に振り，出た目の組み合わせによって役（ポーカーに似ている）を決めていくのです。

今回はこれで目的の役が完成していればスペースキーを押して役の登録を行います。

プレイヤーは役ができていなかった場合やもっと違う役を狙いたい場合はそのうちの2個のサイコロを振り直してもかまいません（というより，第1投の出目を見て役を狙うのですが)。振り直すときはA，Dキーまたはテンキーの4，6でサイコロを選び，リターンキーを押します。

こうして15回役を作ります。といってもひたすら高い役を目指すのがゲームの目的ではありません。KISMETに存在する役は15種類。これらをひとつずつ作っていくのです。サイコロによってできた目はプレイヤーが評価して，どの役を狙ったものかを指定します。15回の試行はどれもなんらかの役に割り当てられなければなりません。うまく役ができなかったときは，どれかの役をつぶさなくてはいけないのです。どの役をどの時点でつぶすかがこのゲームの決め手になってきます。こうして成立した役から得点を合計して勝負が決まるわけです。

役の内容

役は大きく分けてBASICセクションとKISMETセクションの2種類があります。

1)　BASICセクション

ACESからSIXESまであり，その数字のサイコロの目のみ得点となります。

得点＝その役のサイコロの目×個数

となります。つまり，

1，2，1，5，1

という目を“ACES”として登録するのななら得点は3となります。

2)　KISMETセクション

基本的なところはポーカーの役とよく似ています。ただ，サイコロの目の色によって成立する役があります。得点は，

得点＝目の総和＋ボーナス

というのが基本となっています。

では役を解説しましょう。

● TWO PAIR SAME COLOR

同じ色同士のツーペアです。残り1個の目はなんでもかまいません。

● THREE OF A KIND

いわゆるスリーカードです。3個の目が同じなら残りはなんでもかまいません。

● STRAIGHT

いわゆるストレートです。5個の目が続き番号になっていなければなりません。

● FLUSH

いわゆるフラッシュです。同じ色だけでできた手です。

● FULLHOUSE

いわゆるフルハウスです。

● FULLHOUSE SAME COLOR

フルハウスのうち，カードがすべて同じ色の場合です。

● FOUR OF A KIND

いわゆるフォーカードです。

● YARBOROUGH

これはなんでも点になります。目の総和が高いときにでも指定してください。

● KISMET

いわゆるファイブカード。すべてのサイコロの目が同じ場合です。

これらの役を見てわかるように上位の役は下位の役の代わりになります。たとえば，

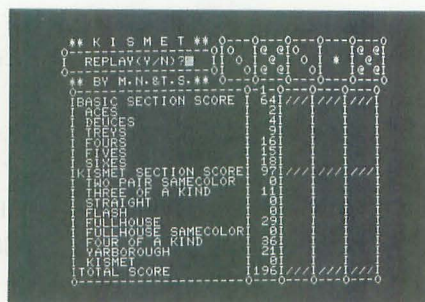
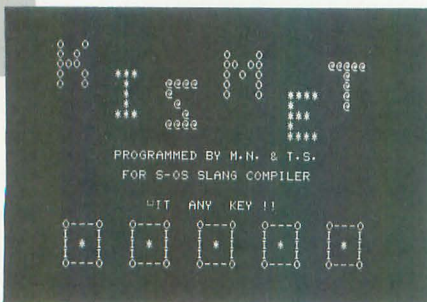
1,1,1,6,6

のようなFULLHOUSE SAME COLORは，TWO PAIR SAME COLOR, THREE OF A KIND, FLUSH, FULLHOUSE, YARBOROUGH, ACES, SIXESのいずれのように指定しても有効となります。

KISMETセクションの得点は，

TWO OF A KIND	合計
THREE OF A KIND	合計
STRAIGHT	30
FLUSH	35
FULLHOUSE	合計＋15
FULLHOUSE SAME COLOR	合計＋20
FOR OF A KIND	合計＋25
YARBOROUGH	合計
KISMET	合計＋50

となります。

BASICセクションでは各々の数字の合計しか得点になりませんが，集計のときにこの部分の合計が63点以上なら35点，71点

以上なら55点の，79点以上なら75点のボーナスがつきます。

注意

SLANGを使っている方ならコンパイル作業は特に説明の必要はないでしょう。SLANGを立ち上げてそのまま，

]CKISMET

のように入力してください。なお，MZ-2500やX1turboの方は忘れずに25行モードにしてから実行してください。

キー操作はカーソル移動にカーソルキー，テンキー，“W, X, A, D”を使用します。

なお，高得点をあげるためにはKISMETセクションにばかり点を集中せずに，BASICセクションで平均3個以上の得点を記録してボーナスを狙う必要が出てきます。

＊　　＊　　＊

色が出ないと寂しいので当初はMAGICを使うつもりでしたが動かなくなる機種が多いのでやめました。同じ色の目には同じキャラクタを使っているのでなんとかわかってもらえるのではないでしょうか。

リスト1

```
  1 /*                             */
  2 /*  KISMET X1.SL               */
  3 /*                             */
  4 /*   for S-OS SLANG COMPILER */
  5 /*   ver. 1.01   '90.08.21    */
  6 /*    T.SAKAKI & M.NAGIRA      */
  7 /*                             */
  8
  9
 10 ARRAY   BYTE DICE[4]=[1,1,1,1,1];
 11 ARRAY   SC[3][14];
 12
 13 VAR  PS_MAX=3,PS=0,YK=0;
 14 VAR  HOLD=0;
 15
 16 CONST   NONSELECT=851;
 17 CONST   BUFF=$E000;
 18 CONST   COLORF=$0026;
 19
 20
 21 CAST_DICE()
 22   VAR I;
 23   BEGIN
 24      FOR I=0 TO 4 [
 25         IF BIT(HOLD,I) == FALSE
 26            DICE[I]=RND(6)+1;
 27      ]
 28      DICE_PRT();
 29   END;
 30
 31
 32 CAST()
 33   VAR I,KEY;
 34   BEGIN
 35      IF HOLD == 00011111B
 36         EXIT;
 37      REPEAT [
 38         KBUF_CLR();
 39         WAIT(5);
 40         CAST_DICE();
 41         IF (KEY=INKEY(0)) == $1B   /* GETKEY */
 42            STOP();
 43      ] UNTIL KEY == ' ';
 44      KBUF_CLR();
 45      FOR I=0 TO 2 [
 46         WAIT(7+I*7);
 47         CAST_DICE();
 48      ]
 49   END;
 50
 51
 52 HOLD_OR_CAST()
 53   VAR I,J,X,Y,XO,IO;
 54   BEGIN
 55      HOLD=00011111B;
 56      I=0; IO=0;
 57      XO=20; Y=1;
 58      KBUF_CLR();
 59      WHILE TRUE [
 60         X=I*4+20;
 61         IF BIT(HOLD,IO) == TRUE [
 62            DICE_SUB(IO,XO,Y);
 63         ] ELSE [
 64            HOLD_CUR(0,XO,Y);
 65         ] ENDIF;
 66            WAIT(10);
 67         HOLD_CUR(1,X,Y);
 68            WAIT(10);
 69         XO=X; IO=I;
 70         CASE INKEY(0) OF [          /* GETKEY */
 71            ' ' :   [
 72               DICE_PRT();
 73               EXIT;
 74            ]
 75            '4','A': [
 76               IF I == 0   I=4;
 77               ELSE  --I;
 78            ]
 79            '6','D': [
 80               IF I == 4   I=0;
 81               ELSE  ++I;
 82            ]
 83            $0D,'5','S':   [
 84               IF BIT(HOLD,I) == FALSE [
 85                  HOLD=SET(HOLD,I);
 86               ] ELSE [
 87                  HOLD=RESET(HOLD,I);
 88               ] ENDIF;
 89               KBUF_CLR();
 90            ]
 91            $1B :    STOP();
 92            'G' :   [
 93               HOLD=0;
 94               FOR J=0 TO 4 [
 95                  HOLD_CUR(0,J*4+20,Y);
 96               ]
 97            ]
 98         ]
 99         KEY_WAIT();
100      ]
101   END;
102
103
104 DICE_PRT()
105   VAR I;
106   BEGIN
107      FOR I=0 TO 4 [
108         DICE_SUB(I,20+I*4,1);
109      ]
110   END;
111
112
113 DICE_SUB(I,X,Y)
114   BEGIN
115      CASE DICE[I] OF [
116         1 :    [
117         /* COLOR(1);    */
118            LOCATE(X,Y);    PRINT("   ");
119            LOCATE(X,Y+1);  PRINT(" * ");
120            LOCATE(X,Y+2);  PRINT("   ");
121         ]
122         2 :    [
123         /* COLOR(2);    */
124            LOCATE(X,Y);    PRINT("@  ");
125            LOCATE(X,Y+1);  PRINT("   ");
126            LOCATE(X,Y+2);  PRINT("  @");
127         ]
128         3 :    [
129         /* COLOR(4);    */
130            LOCATE(X,Y);    PRINT("O  ");
131            LOCATE(X,Y+1);  PRINT(" O ");
132            LOCATE(X,Y+2);  PRINT("  O");
133         ]
134         4 :    [
135         /* COLOR(4);    */
136            LOCATE(X,Y);    PRINT("O O");
137            LOCATE(X,Y+1);  PRINT("   ");
138            LOCATE(X,Y+2);  PRINT("O O");
139         ]
140         5 :    [
141         /* COLOR(2);    */
142            LOCATE(X,Y);    PRINT("@ @");
143            LOCATE(X,Y+1);  PRINT(" @ ");
144            LOCATE(X,Y+2);  PRINT("@ @");
145         ]
146         6 :    [
147         /* COLOR(1);    */
148            LOCATE(X,Y);    PRINT("* *");
149            LOCATE(X,Y+1);  PRINT("* *");
150            LOCATE(X,Y+2);  PRINT("* *");
151         ]
152      ]
153   /* COLOR(7);    */
154   END;
155
156
157 HOLD_CUR(I,X,Y)
158   BEGIN
159      CASE I OF [
160         0 :    [
161            LOCATE(X,Y);    PRINT("   ");
162            LOCATE(X,Y+1);  PRINT("   ");
163            LOCATE(X,Y+2);  PRINT("   ");
164         ]
165         1:     [
166            LOCATE(X,Y);    PRINT(" ? ");
167            LOCATE(X,Y+1);  PRINT("? ?");
168            LOCATE(X,Y+2);  PRINT(" ? ");
169         ]
170      ]
171   END;
172
173
174 MAIN()
175   VAR TURN;
176   BEGIN
177      WHILE TRUE [
178         INIT();
179         TITLE();
180         INTGET();
181
182         REPEAT [
183            INIT_SC();
184            BG_PRT();
185            HIT_SPC();
186            FOR TURN=1 TO 15 [
187               FOR PS=0 TO PS_MAX [
188                  HOLD=0;
189                  NAME();
190                  LOCATE(24+PS*4,5);
191                     PRINT("*");
192                  CAST();
193                  HOLD_OR_CAST();
194                  CAST();
195                  LOCATE(1,2);
196                     PRINT(" CAST ONCE MORE! ");
197                  HOLD_OR_CAST();
198                  CAST();
199                  IF TURN != 15 [
200                     SELECT();
201                  ] ELSE [
202                     LAST1();
203                     HIT_SPC();
204                  ] ENDIF;
```

```
205                 LOCATE(24+PS*4,5);
206                    PRINT(PS+1);
207             ]
208          ]
209
210          TOTAL();
211       ] UNTIL REPLAY();
212    ]
213   END;
214
215
216 KBUF_CLR()
217   BEGIN
218     WHILE INKEY(0) != 0 [          /* GETKEY */
219     ]
220   END;
221
222
223 WAIT(I)
224   VAR WT,KEY;
225   BEGIN
226      WT=100;
227      I=I*WT;
228      WHILE I-- != 0 [
229         IF (KEY=INKEY(0)) != 0 [    /* GETKEY */
230            I=0;
231         ] ENDIF;
232      ]
233   END(KEY);
234
235
236 SELECT()
237   VAR Y,YO,KEY,YKO;
238   BEGIN
239      YK=0; YKO=14:
240      Y=7;  YO=22;
241      WHILE TRUE [
242         LOCATE(1,2);
243            PRINT(" * S E L E C T * ");
244         IF Y != YO [
245            LOCATE(2,YO);  PRINT(" ");
246            IF SC[PS][YKO] == NONSELECT [
247               LOCATE(23+PS*4,YO); PRINT("    ");
248            ] ENDIF;
249            IF SC[PS][YK] == NONSELECT [
250               LOCATE(2,Y);  PRINT(">");
251               LOCATE(23+PS*4,Y);  PRINT("---");
252            ] ELSE [
253               LOCATE(2,Y);  PRINT("*");
254            ] ENDIF;
255            YO=Y; YKO=YK;
256         ] ENDIF;
257         KEY=INKEY(0);              /* GETKEY */
258         KEY_WAIT();
259         CASE KEY OF [
260            $0D :     [
261               IF SC[PS][YK] == NONSELECT [
262                  JUDGE();
263                  LOCATE(23+PS*4,Y);
264                     PRINT(FORM$(SC[PS][YK],3));
265                  LOCATE(2 ,Y);  PRINT(" ");
266                  IF HIT_SPC() == 'C' [
267                     LOCATE(23+PS*4,Y);
268                        PRINT("---");
269                     LOCATE(2 ,Y);
270                        PRINT(">");
271                     SC[PS][YK]=NONSELECT;
272                  ] ELSE [
273                  EXIT FROM FUNC;
274                  ] ENDIF;
275               ] ENDIF;
276            ]
277            '2','X': [
278               IF Y == 12 [
279                  Y=14; YK=6;
280               ] EF Y == 22 [
281                  Y=7;  YK=0;
282               ] ELSE [
283                  ++Y;  ++YK;
284               ] ENDIF;
285            ]
286            '8','W': [
287               IF Y == 14 [
288                  Y=12; YK=5;
289               ] EF Y == 7 [
290                  Y=22; YK=14;
291               ] ELSE [
292                  --Y;  --YK;
293               ] ENDIF;
294            ]
295            $1B :    STOP();
296         ]
297      ]
298   END;
299
300
301 INIT_SC()
302   VAR I,J;
303   BEGIN
304      PRINT("¥C");
305      FOR I=0 TO PS_MAX [
306         FOR J=0 TO 14 [
307            SC[I][J]=NONSELECT;
308         ]
309      ]
310   END;
311
312
313 KEY_WAIT()
314   VAR I,J,K_WT1,K_WT2;
315   BEGIN
316      K_WT1=1500;
317      K_WT2=1500;
318      FOR I=0 TO K_WT1 [
319         IF INKEY(0) == 0 [          /* GETKEY */
320            FOR J=0 TO K_WT2 [
321            ]
322            EXIT FROM FUNC;
323         ] ENDIF;
324      ]
325   END;
326
327
328 BG_PRT()
329   VAR I;
330   BEGIN
331      LOCATE(0,0);
332         PRINT(" ** K I S M E T ** ");
333      LOCATE(0,4);
334         PRINT(" ** BY M.N.&T.S.** ");
335      LOCATE(0,6);
336         PRINT("  BASIC SECTION SCORE¥N");
337         PRINT("   ACES¥N");
338         PRINT("   DEUCES¥N");
339         PRINT("   TREYS¥N");
340         PRINT("   FOURS¥N");
341         PRINT("   FIVES¥N");
342         PRINT("   SIXES¥N");
343         PRINT("  KISMET SECTION SCORE¥N");
344         PRINT("   TWO PAIR SAMECOLOR¥N");
345         PRINT("   THREE OF A KIND¥N");
346         PRINT("   STRAIGHT¥N");
347         PRINT("   FLASH¥N");
348         PRINT("   FULLHOUSE¥N");
349         PRINT("   FULLHOUSE SAMECOLOR¥N");
350         PRINT("   FOUR OF A KIND¥N");
351         PRINT("   YARBOROUGH¥N");
352         PRINT("   KISMET¥N");
353         PRINT("  TOTAL SCORE");
354
355      BOX_PRT(0,1,18,3);
356      FOR I=0 TO 4 [
357         BOX_PRT(19+I*4,0,23+I*4,4);
358      ]
359      BOX_PRT(1,5,22,24);
360      FOR I=0 TO 3 [
361         BOX_PRT(22+I*4,5,26+I*4,24);
362         LOCATE(23+I*4, 6);   PRINT("///");
363         LOCATE(23+I*4,13);   PRINT("///");
364         LOCATE(23+I*4,23);   PRINT("///");
365      ]
366      FOR I=0 TO PS_MAX [
367         LOCATE(24+I*4,5); PRINT(I+1);
368      ]
369
370      DICE_PRT();
371   END;
372
373
374 BOX_PRT(X,Y,XE,YE)
375   VAR I;
376   BEGIN
377      LOCATE(X ,Y);  PRINT("O");
378      FOR I=X+1 TO XE-1 [ PRINT("-"); ]
379      LOCATE(XE,Y);  PRINT("O");
380
381      FOR I=Y+1 TO YE-1 [
382         LOCATE(X ,I);      PRINT("I");
383         LOCATE(XE,I);      PRINT("I");
384      ]
385
386      LOCATE(X ,YE); PRINT("O");
387      FOR I=X+1 TO XE-1 [ PRINT("-"); ]
388      LOCATE(XE,YE); PRINT("O");
389   END;
390
391
392 SN_CHK(I)
393   VAR J,K;
394   BEGIN
395      K=0;
396      FOR J=0 TO 4 [
397         IF DICE[J] == DICE[I] [
398            ++K;
399         ] ENDIF;
400      ]
401   END(K);
402
403
404 SC_NUM(I)
405   VAR J,K;
406   BEGIN
407      K=0;
408      FOR J=0 TO 4 [
409         IF DICE[J] == DICE[I] [
410            ++K;
411         ] EF (DICE[J] + DICE[I]) == 7 [
412            ++K;
413         ] ENDIF;
414      ]
415   END(K);
416
417
418 ALL_SUM()
419   VAR I,SUM;
420   BEGIN
421      SUM=0;
422      FOR I=0 TO 4 [
423         SUM=SUM+DICE[I];
424      ]
425   END(SUM);
426
427
428 JUDGE()
429   VAR I,J;
430   BEGIN
431      SC[PS][YK]=0;
432
433      CASE YK OF [
434         00 TO 05 :  [      /* BASIC SECTION */
435            FOR I=0 TO 4 [
436               IF DICE[I] == YK+1 [
437                  SC[PS][YK]=SC[PS][YK]+YK+1;
438               ] ENDIF;
439            ]
440         ]
441         06  :       [      /* TWO PAIR SAMECOLOR */
442            SORT();
443            CASE SC_NUM(1) OF [
444               01 TO 03 :  EXIT;
445               04  :    [
446                  IF SN_CHK(1) == 1 [
447                     EXIT;
448                  ] EF SN_CHK(1) == 3 [
449                     EXIT;
450                  ] ENDIF;
451                  SC[PS][YK]=ALL_SUM();
452               ]
453               05  :    [
454                  SC[PS][YK]=ALL_SUM();
455               ]
456            ]
```

▶マシン語カクテルを見るとペンギン，電脳倶楽部では犬，という西川さんの正体は（音楽が得意そうなので），昔テレビのコマーシャルで出ていたヘッドホンをつけていたサルでしょ？　あ，おこらないで。
水上　晴雄(21)東京都

```
457         ]
458         07  :       [     /* THREE OF A KIND */
459            SORT();
460            IF SN_CHK(2) >= 3 [
461               SC[PS][YK]=ALL_SUM();
462            ] ENDIF;
463         ]
464         08  :       [     /* STRAIGHT */
465            SORT();
466            J=1;
467            FOR I=0 TO 3 [
468               J=J*(DICE[I+1]-DICE[I]);
469            ]
470            IF J == 1 [
471               SC[PS][YK]=30;
472            ] ENDIF;
473         ]
474         09   :      [     /* FLASH */
475            IF SC_NUM(0) == 5 [
476               SC[PS][YK]=35;
477            ] ENDIF;
478         ]
479         10   :      [     /* FULLHOUSE */
480            SORT();
481            I=SN_CHK(0);
482            J=SN_CHK(4);
483            IF (I*J) == 6 [
484               SC[PS][YK]=ALL_SUM()+15;
485            ] EF I == 5 [
486               SC[PS][YK]=ALL_SUM()+15;
487            ] ENDIF;
488         ]
489         11   :      [     /* FULLHOUSE SAMECOLOR */
490            SORT();
491            IF SC_NUM(0) != 5 [
492               EXIT;
493            ] ENDIF;
494            I=SN_CHK(0);
495            J=SN_CHK(4);
496            IF (I*J) == 6 [
497          -    SC[PS][YK]=ALL_SUM()+20;
498            ] EF I == 5 [
499               SC[PS][YK]=ALL_SUM()+20;
500            ] ENDIF;
501         ]
502         12  :       [     /* FOUR OF A KIND */
503            SORT();
504            IF SN_CHK(2) >= 4
505               SC[PS][YK]=ALL_SUM()+25;
506         ]
507         13  :       [     /* YARBOROUGH */
508            SC[PS][YK]=ALL_SUM();
509         ]
510         14  :       [     /* KISMET! */
511            IF SN_CHK(0) == 5
512               SC[PS][YK]=ALL_SUM()+50;
513         ]
514      ]
515   END;
516
517
518 SORT()
519   VAR GAP,I,J;
520   BEGIN
521      GAP=3;
522      FOR I=0 TO 2 [
523         FOR J=0 TO 4-GAP [
524            IF DICE[J] > DICE[J+GAP]
525               SWAP(J,J+GAP);
526         ]
527         --GAP;
528      ]
529   END;
530
531
532 SWAP(I,J)
533   VAR TEMP;
534   BEGIN
535      TEMP=DICE[I];
536      DICE[I]=DICE[J];
537      DICE[J]=TEMP;
538   END;
539
540
541 TOTAL()
542   VAR B_SC,K_SC,T_SC,BONUS;
543   BEGIN
544      FOR PS=0 TO PS_MAX [
545
546         B_SC=0;
547         K_SC=0;
548         T_SC=0;
549         BONUS=0;
550
551         KBUF_CLR();
552         NAME();
553            WAIT(150);
554         FOR YK=0 TO 5 [
555            B_SC=B_SC+SC[PS][YK];
556            LOCATE(23+PS*4,6);
557               PRINT(FORM$(B_SC,3));
558         ]
559            WAIT(150);
560         FOR YK=6 TO 14 [
561            K_SC=K_SC+SC[PS][YK];
562            LOCATE(23+PS*4,13);
563               PRINT(FORM$(K_SC,3));
564         ]
565            WAIT(150);
566         IF B_SC > 62   BONUS=35;
567         IF B_SC > 70   BONUS=55;
568         IF B_SC > 78   BONUS=75;
569         T_SC=B_SC+K_SC+BONUS;
570         LOCATE(23+PS*4,23);
571            PRINT(FORM$(T_SC,3));
572         ]
573    END;
574
575
576 HIT_SPC()
577   VAR KEY;
578   BEGIN
579      REPEAT [
580         KBUF_CLR();
581         LOCATE(1,2);   PRINT(" HIT SPACE KEY ! ");
582         LOCATE(1,2);   PRINT(SPC$(17));
583         IF (KEY=INKEY(0)) == $1B      /* GETKEY */
584            STOP();
585      ] UNTIL (KEY == ' ') OR (KEY == 'C');
586   END(KEY);
587
588
589 HIT_ANY(X,Y)
590   VAR KEY;
591   BEGIN
592      REPEAT [
593         KBUF_CLR();
594         LOCATE(X,Y);   PRINT(" HIT  ANY  KEY !!");
595         LOCATE(X,Y);   PRINT(SPC$(17));
596         IF (KEY=INKEY(0)) == $1B      /* GETKEY */
597            STOP();
598      ] UNTIL KEY != $00;
599   END;
600
601
602 REPLAY()
603   VAR I;
604   BEGIN
605      LOCATE(1,2);   PRINT("  REPLAY(Y/N)?   ");
606      KBUF_CLR();
607      LOCATE(15,2);
608      WHILE TRUE [
609         CASE INKEY(1) OF [
610            'Y',' ',$0D :  [
611                     I=FALSE; EXIT;
612            ]
613            'N'         :  [
614                     I=TRUE;  EXIT;
615            ]
616            $1B         :  STOP();
617         ]
618      ]
619   END(I);
620
621
622 TITLE()
623   VAR I,J;
624   BEGIN
625      LOCATE(0,1);
626      PRINT(" O  O                                    ");
627      PRINT(" O O                  O   O              ");
628      PRINT(" OOO                  OO OO        @@@@@ ");
629      PRINT(" O  O   ***           O O O          @   ");
630      PRINT(" O  O    *    @@@@    O   O          @   ");
631      PRINT("         *    @       O   O   ****   @   ");
632      PRINT("         *     @              *      @   ");
633      PRINT("        ***     @             ***        ");
634      PRINT("              @@@@            *          ");
635      PRINT("                              ****       ");
636      LOCATE(0,12);
637      PRINT("        PROGRAMMED BY M.N. & T.S.¥N");
638      PRINT("¥N");
639      PRINT("         FOR S-OS SLANG COMPILER¥N");
640      FOR I=0 TO 4 [
641         BOX_PRT(2+I*8,19,6+I*8,23);
642         DICE_SUB(I,3+I*8,20);
643      ]
644      HIT_ANY(11,17);
645      KBUF_CLR();
646      FOR J=0 TO 6 [
647         FOR I=0 TO 4 [
648            DICE[I]=RND(6)+1;
649            DICE_SUB(I,3+I*8,20);
650         ]
651         WAIT(10);
652      ]
653      WAIT(50);
654   END;
655
656
657 INTGET()
658   VAR I,KEY;
659   BEGIN
660      FOR I=0 TO 7 [
661         LOCATE(1,16+I);   PRINT(SPC$(38));
662      ]
663      BOX_PRT(02,16,37,23);
664
665      LOCATE(04,17); PRINT("HOW MANY PLAYERS?(1-4)");
666      WHILE TRUE [
667         LOCATE(26,17);
668            PRINT(SPC$(5));
669         LOCATE(26,17);
670            KEY=INKEY(1);
671            PRINT(KEY-'0');
672         IF KEY<'5' AND KEY>'0' [
673            PS_MAX=KEY-'0'-1;
674            EXIT;
675         ] EF KEY == $1B [
676           STOP();
677         ] ENDIF;
678      ]
679
680      FOR PS=0 TO PS_MAX [
681         LOCATE(04,19+PS);
682            PRINT("INPUT YOUR NAME,PLAYER");
683            PRINT(PS+1);   PRINT(":");
684            LINPUT(BUFF+PS*$10,9);
685
686         IF MEMW[BUFF+PS*$10] == $2020 [
687            CASE PS OF [
688               00 :  MEMW[BUFF+$00]=$3158;
689               01 :  MEMW[BUFF+$10]=$5A4D;
690               02 :  [
691                  MEMW[BUFF+$20]=$484F;
692                  MEMW[BUFF+$22]=$5821;
693               ]
694               03 :  [
695                  MEMW[BUFF+$30]=$3658;
696                  MEMW[BUFF+$32]=$4B38;
697               ]
698            ]
699         ] ENDIF;
700      ]
701   END;
702
703
704 NAME()
705   BEGIN
706      LOCATE(1,2);
707      PRINT("PLAYER");
708      PRINT(PS+1);   PRINT(":");
```

▶電脳倶楽部のボールペン，ありがとうございました。さっそく砂時計の時間を計ったところ，約3分10秒でした。カップラーメンを食べるとき（ほとんどないが）にでも使わせていただきます。もちろん，ボールペンとしても使いますけど。　松本　太(20)大阪府

```
709        PRINT(MSX$(BUFF+PS*$10));
710     END;
711
712
713 INIT()
714     BEGIN
715        WIDTH(40);
716        PRINT("¥C");
717     /* INIT_JS();  */
718     END;
719
720
721 LAST1()
722     VAR I,Y;
723     BEGIN
724        FOR I=0 TO 14 [
725           IF SC[PS][I] == NONSELECT [
726              YK=I;
727              Y=I+7+(YK>5);
728              JUDGE();
729              LOCATE(23+PS*4,Y);
730                 PRINT(FORM$(SC[PS][YK],3));
731              EXIT FROM FUNC;
732           ] ENDIF;
733        ]
734     END;
735
736
737 /*    ORIGINAL PROGRAMMED BY M.NAGUIRA
738 /*       FOR CZ-8FB01 V1.0
739 /*    S-OS VERSION PROGRAMMED BY T.SAKAKI
740 /*       FOR SLANG COMPILER
741 /*       MACHINE:CZ-851CR,CZ-880CB
742 /*    SPECIAL THANKS TO MR.ASO
```

全機種共通システムインデックス

■85年6月号
序論　共通化の試み
第1部　S-OS"MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年9月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法〈1〉
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法〈3〉
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法〈4〉
■87年2月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアベ作成ツール CONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年1月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲーム ELFES
■88年3月号
第60部　構造型コンパイラ言語 SLANG
■88年4月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年5月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年7月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年8月号
第68部　マルチウィンドウエディタ WINER
■88年9月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年1月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年2月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"〈再掲載〉
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年4月号
第79部　SLANG 用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータ RING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲーム　TICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85年　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム　PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ DIO. LIB
■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム　SQUASH！
第95部　X68000対応S-OS"SWORD"
特別付録　PC-286対応S-OS"SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ
■91年1月号
第102部　ブロックアクションゲーム COLUMNS

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

ハードウェア工作入門《8》

センサー回路その2

Misawa Kazuhiko
三沢 和彦

A/Dコンバータを利用したセンサー回路の入門ですが，今回は，そこで使われるオペアンプを詳しく取り上げてみましょう。アナログ信号の増幅ということでちょっとデジタル回路から離れますが，利用度の高いICですからぜひ理解してください。

前回はセンサー理論編ということで，A/Dコンバータに外付けするためのセンサー回路を設計するための予備知識について解説しました。実習回路例として，アルコールセンサーと光センサーとを挙げましたが，光センサーの回路では皆さんにはあまり馴染みのないオペアンプというICを使いました。実際，オペアンプはTTL-ICに並んで最もよく使われているICのシリーズなのですが，アナログ信号用のICということで，これまでパソコン雑誌上で解説される機会は多くなかったように思います。そこで今月はもう１回理論編ということにして，オペアンプの使い方について徹底的に学んでみたいと思います。

アナログ信号の増幅

アナログ信号がデジタル信号と異なる大きな点は連続した値を取るということです。10月号の解説を読み返してみるとわかるとおり，A/Dコンバータには分解能というものがあります。たとえば，フルスケール5Vの８ビットA/Dコンバータの最小単位は5/256＝0.01953125Vということになります。ところが，アナログ量は連続量なので，もちろん0.01Vという量も存在します。0.01VがA/Dコンバータに入力されたとしても分解能の5/256Vよりも小さいためにA/D変換されたあとは0Vということになってしまいます。0.01Vでも0.011Vでも0.019Vであっても，そして本当に0Vのときも含めてすべて0Vになってしまいます。

そのようなデジタル量における誤差（量子誤差という）はアナログ量を扱うセンサー回路において重大な問題となります。なぜなら，センサーが自然界の物理量を検出したときに，そのセンサーが感度の高い優れたものであるほど，出力信号の大きさは微小なことが多いからです。

先月の光センサー回路がそのよい例で，フォトダイオードの電気的な信号出力をそのままA/Dコンバータに入れてもほとんど分解能以下になってしまうのです。そのままではすべて0Vになって話になりません。したがって，光センサーで検出した光の強度をA/Dコンバータを通してパソコンで処理しようとすれば，A/Dコンバータのフルスケール5V近くまで信号を強めてやらなければなりません。しかも，センサーの出力はアナログ量ですから，強めた信号もそれと同じようにアナログ量でなければ，都合が悪いのです。

そこで，あるアナログ信号を入力させて，その定数倍（倍率という）の信号を出力するような回路が便利です。その回路をアナログ増幅器というのです。たとえば，倍率100倍の増幅器では，0.01Vは１Vに，0.011Vは1.1Vに，そして0.019Vは1.9Vになるので，分解能5/256VのA/Dコンバータではその３つは区別できるのです。

パソコンの内部では，増幅器はそんなに多く使われていないように思うでしょうが，FM音源やAD PCM，CRTディスプレイなどのまわりは増幅器が必ず使われています。もちろん，テレビやビデオ，オーディオ等の機器では，ふたを開けるとどこもかしこも増幅器だらけです。それだけ，アナログ信号の増幅というのは電子回路において大切な働きなのです。

図１　オペアンプの基本的な使い方

1）反転増幅器　　2）非反転増幅器

アナログ回路の万能選手＝オペアンプ

オペアンプは演算増幅器（Oprational Amplifier）の略で，アナログ信号の増幅器としての基本的な機能をひとまとめにしたICです。増幅度の設定のために若干の抵抗器などを外付けにする必要がありますが，デジタル回路におけるTTL-ICと同じように中身はブラックボックスとして，ただ端子どうしをつないでいくだけで大部分の回路は組めてしまうものです。オペアンプを使った回路図を見ると，三角形とそのなかに＋と−の記号しかありません。中身は複雑なICでも機能としてはそれだけ単純明快なのです。

さて，オペアンプの初歩的な使い方には，

1）　反転増幅器　（図１－１）
2）　非反転増幅器（図１－２）

の２種類があります。先月載せた光センサーの回路にある２段のオペアンプもこれらの使い方をしているにすぎません。そこで今回はこの２つに限定して説明します。

1）　反転増幅器

回路図の入力端子にE_iの電圧をかけたときに出力E_oは

$E_o=-(R_f/R_i)E_i$

となります。このとき，入力電圧には正負両方がかけられ，また出力電圧も正負両方がありえます。したがって，倍率$-(R_f/R_i)$が負ですから，入力が正のときは出力が負，入力が負のときは出力が正というように「反転」します。これが反転増幅器という意味です。電圧の絶対値の倍率は外付けのR_fとR_iの値を適当に選ぶことによって，自由に設定できます。もちろんR_fをR_iより小さくして倍率を１より小さくすることも可能で，この場合は増幅器というよりは減衰器ということになります。

2）　非反転増幅器

回路図の入力端子にE_iの電圧をかけたときに出力E_oは，

$E_o = (1 + R_f/R_i)\ E_i$

となります。「非反転」の意味はもうおわかりでしょう。倍率が正なので，出力電圧の符号は入力電圧の符号に対して反転しないからです。また，非反転増幅器の場合はどのように抵抗値を設定しても倍率は1よりも大きい点も異なります。

オペアンプの代表的な2種類の回路について，以上のことを理解するだけで簡単に使いこなすことができるのです。しかし，もう少し一般的なオペアンプの仕組みを理解すると，オペアンプと外付け抵抗を使ったときになぜ上のような倍率が得られるのかということを2つまとめて説明することができます。興味のある人は囲み記事のほうを参考にしてください。

オペアンプの使用例

では，先月号に載せた光センサー回路でのオペアンプの使い方を今述べた2つの基本回路にしたがって考えてみましょう。図2がその回路図です。オペアンプが2段ありますが，1段目は反転増幅器と同じような使い方で，電流増幅器といいます。これは，フォトダイオードの出力信号の電流を電圧に変える増幅器です。2段目は上で述べた非反転増幅器とまったく同じです。

1段目の回路を図1-1の反転増幅器と見比べてください。入力電圧のところにフォトダイオードがつながっているのはわかりますが，入力側の抵抗R_iがありません。R_iがないとすると，倍率$-(R_f/R_i)$を考えると出力E_oは無限大になってしまうように思えますが，実際はフォトダイオードに内部抵抗があり，ちょうど$I=E_i/R_i$に相当する電流を流すのです。したがって，フォトダイオードの出力電流がIのときに出力E_oは，

$E_o = R_f I$

となります。このように，信号源が一定電流を流してくれるような場合には，反転増幅器の入力抵抗を省略した形の回路を使います。このような増幅器を電流増幅器というのです。このとき，フォトダイオードの電流の向きは出力電圧が正になるように接続します。回路図では，$R_f = 1M\Omega$を選んでいますが，フォトダイオードの出力電流が$1\mu A$程度なので（もちろん光強度による），1段目の電流増幅器の出力は1V程度ということになります。

1MΩと並列に入っているコンデンサは入力信号が急激に変化したときにある程度貯め込んでから出力するためのもので，正確には積分回路といいますが，今回は入門編ということで詳しい説明は省略することにします。

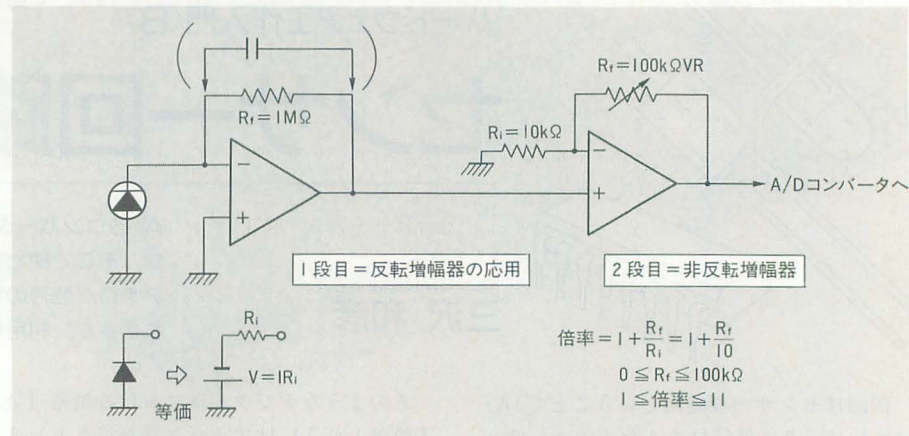



図2　光センサー回路

2段目は非反転増幅器そのものなので，問題はないと思いますが，R_iが10kΩの固定抵抗に対して，R_fは100kΩの可変抵抗にしてあります。倍率は$(1+R_f/R_i)$で与えられますから，可変抵抗を調整することによって，1倍から11倍まで変えることができるようになっています。これは，だいたい1段目の出力が1Vであることから，2段目の出力（A/Dコンバータの入力）を5Vに設定するためにあと5倍程度の増幅器が必要なのですが，使用する条件によって光の強度が変わることを予想して，倍率を変化させられるように便宜を図ったのです。

LM358と各種オペアンプについて

以上でひと通りオペアンプの使い方を実際の回路例も含めて説明しましたが，使っている部品の仕様については何も述べませんでした。オペアンプと一口にいっても極めてたくさんの型番があり，いざ部品を選ぶにあたって迷ってしまうこともあるくらいです。いずれは皆さんが自分で回路設計することもあると思いますので，代表的な個々のオペアンプについて，部品の選び方を説明しておきます。

1）汎用オペアンプ

特別な回路でない限り，まず最初に候補に挙がるのが741シリーズと呼ばれるものです。上で述べた，反転・非反転増幅器の使い方をする限りこの741で十分です。ただし，最近この741は廃品種になった模様で，その代わりに4558シリーズが最もポピュラーなオペアンプになっています。しかし，いくら汎用オペアンプといってもこの741(4558)は万能ではありません。特に以下に述べる点で注意が必要なときはより進んだオペアンプを使用します。

2）単電源オペアンプ

今回のフォトダイオードの増幅回路でこの741タイプを使用しなかった最大の理由は，741が±12Vの電源を必要とする点です。パソコンではTTL+5Vのみの電源が都合がよいのです。では，741を無理に単電源で動作させたらどうなるかというと，まったく使いものにならないといってもよいぐらいです。

そこで，+5Vだけでも使える専用オペアンプが必要になります。今回使用したLM358はその単電源オペアンプの代表です。あるいはLM324のほうがポピュラーともいえるでしょうが，LM358とLM324の違いは，同等なオペアンプが2個入りか4個入りかの違いだけです。また，1回路当たりの性能は741と同等と考えてかまいません。

3）FET入力オペアンプ

オペアンプを微弱信号の増幅に使用したときに問題になるのが，オペアンプの理想特性（理想特性については囲み記事参照）からのずれです。特に，フォトダイオードのような電流信号出力のセンサーを使う場合，オペアンプの入力端子に流れ込む「バイアス電流」というものの大きさが問題になります。FET入力オペアンプはこのバイアス電流を小さくし，微弱電流の増幅に特に高性能を示すものです。代表的なのはLF356です。

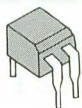
このような事情を考えると，今回の光センサーの増幅回路には本来はこのFET入力オペアンプを使用すべきです。しかし，今回はあまり微小な光強度での精度は必要としないことと，LF356も正負両方の電源が必要なので，単電源FET入力オペアンプを選ぶとかなり特殊になってしまうことから，目をつぶりました。

4）高精度オペアンプ

オペアンプの理想特性からのずれとして次に挙げられるのが，「オフセット電圧」と呼ばれるもので，これはオペアンプの＋入力端子と－入力端子との電位差のことをいいます。理想特性ではこの電位差は0と置いています。問題は，このオフセット電圧が時間的に変動（ドリフト）したときで，そうなると増幅器の倍率も揺らいでしまい精度ががた落ちということになってしまいます。高精度オペアンプOP07はオフセット電圧のドリフトが小さく，またノイズにも強くて，実に幅広く用いられている高精度オペアンプです。

5）高速オペアンプ

最後の問題点は，オペアンプの周波数特性です。オーディオ信号のように時間的に変動する信号を増幅する場合に，オペアンプの反応が追いつくかどうかというのはかなり問題になる点です。あまりに変動が速すぎるとオペアンプの増幅が追いつかなくなり，入力した信号に比例する出力が得られなくなります。特にオーディオ用の増幅器の場合には，音が歪むので使いものになりません。

たとえば741シリーズを使って100倍の倍率を取る場合，歪みを1％に抑えるためには入力を1kHz＝1000Hzにしなければいけません。オーケストラの楽曲は10～20kHzは出ているので，せっかくの名曲も台なしです。このような場合には，専用の高速オペアンプを使うしかありません。今ではオーディオ用オペアンプという名で多数開発されていて，それらは100kHz程度まで倍率の変動が十分抑えられているほか，倍率の変動には現れなくても音の響き具合に影響のある位相歪みというものに対しても満足の性能が出るようになっています。

アナログ回路もだいじょーヴイ！

Oh!X誌上で私の知る限り初めてのアナログ回路入門ということで，ハードウェア工作入門としては少し余計なくらいまで説明してみました。「デジタル回路だけでも全然理解できないのにさらに難しかった」という感想を持つ読者も多いかもしれませんが，何も苦手意識を持つ必要はありません。デジタル回路もアナログ回路も入門のうちは，市販のICをどう組み合わせてつなぐか，というパズルにすぎないのです。

ハードウェアの理解には皆さんがBASICやCの文法を覚える以上に難しいことは何もなく，大切なのは，まず苦手意識を忘れること。その次は，目的の回路に使われている主要なICの役割をたとえチンプンカンプンでもいいから，数多く眺めてみることです。ICの規格表や，回路の実例集などをぱらぱらとめくっているうちに似たようなICの使い方に何度か出会い，知らず知らずのうちに回路の組み方がわかってくるものなのです。

それでは，来月からまた改めてハードウェア工作実習に一緒にチャレンジしようではありませんか。

オペアンプの理想特性

オペアンプをよほど専門的に使わない限りは次に述べる理想特性というものを理解しておけばほぼどんな回路にでも対応できます。

その前に，オームの法則について復習しておきましょう（図3）。オームの法則というのは，抵抗Rの両端に電圧Vをかけたときにその抵抗に流れる電流Iについての関係が，

$V=I\cdot R$

となる法則です。抵抗R_1とR_2が直列になっているとき，全体の抵抗Rは$R=R_1+R_2$になるので，電流Iは，

$I=V/(R_1+R_2)$

になります。またその場合に，R_1，R_2それぞれの両端の電圧はやはりオームの法則にしたがい，

$V_1=IR_1=VR_1/(R_1+R_2)$

$V_2=IR_2=VR_2/(R_1+R_2)$

となります。これはちょうど全体の電圧VをR_1：R_2に内分した値になっていて，簡易ジョイスティックのところで説明した電圧入力の求め方と同じです。そして，オームの法則の成り立つ抵抗の両端で，電流の流れる方向に沿って電位は高いという約束になっています。

では，オームの法則が頭に入ったところで，オペアンプの本質ともいえる理想特性の説明に入りましょう。オペアンプの記号は図1のような三角形で，－入力と＋入力と出力の3端子を持っています。このオペアンプの－入力の電位をE_-，＋入力の電位をE_+とおきます。

さて，通常のオペアンプの使い方は，出力端子から抵抗を通して－入力につないであり，これを負帰還といいます。図1の反転増幅器，非反転増幅器どちらも負帰還がかかっているのがわかるでしょう。負帰還がかかっているときのオペアンプの理想特性というのは，

- **$E_+=E_-$**
- **－入力，＋入力ともに電流は流れ込まない**
- **出力電流はいくらでも流せる**

以上の3つを考えます。

この理想特性を反転，非反転増幅器それぞれの場合に適用させてみます。

1）反転増幅器

図1-1を見てください。

・まず＋入力端子がGNDに落ちているので，

$E_+=0$

・理想特性より，

$E_+=E_-=0$

・出力端子からR_fに流れる電流をIとおくとオームの法則より

$E_o-E_-=E_o=IR_f$

・－入力端子には電流は流れ込まないので，R_iに流れる電流もIだから，

$E_--E_i=-E_i=IR_i$

・Iを消去して，

$E_o=-(R_f/R_i)E_i$

この式が本文中で示した倍率の式です。

2）非反転増幅器

図1-2を見てください。

・まず＋入力端子に入力電圧がそのままかかっているので，

$E_+=E_i$

・理想特性より，

$E_+=E_-=E_i$

・出力端子からR_fに流れる電流をIとおくとオームの法則より，

$E_o-E_-=E_o-E_i=IR_f$

・－入力端子には電流は流れ込まないので，R_iに流れる電流もIだから，

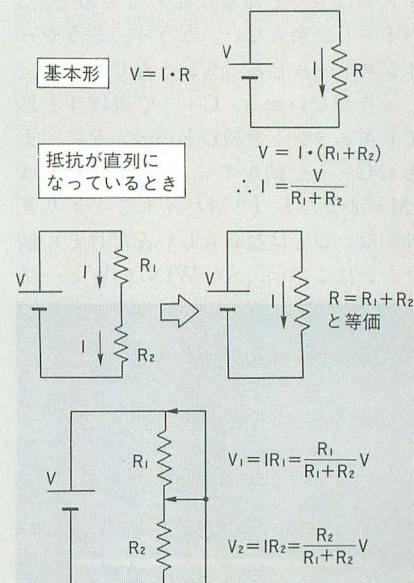
$E_-=E_i=IR_i$

・Iを消去して，

$E_o=(1+R_f/R_i)E_i$

この式が本文中で示した倍率の式です。

図3　オームの法則

★(で)のショートプロぱーてぃ その17

行け行けユーティリティ

Komura Satoshi 古村 聡

今月の(で)のショートプロぱーてぃはツール2本立て。どちらもX68000用で，1本目が英字の字間を詰めてくれるX-BASIC用関数「p_symbol( )」，2本目が少し便利な機能つきのOPMファイル演奏バッチファイル「play.bat」です。

illustration : T.Takahashi

あ、みなさまとっくに明けちゃいましたけどおめでとうございます。(で)でございます。

TEXってのがあるんですけどご存じですか？　このTEXっていうのは大雑把にいうと，ワープロの印刷部分だけ持ってきて強化したようなもんで，テキストファイルに書体やら字の置き方(TEXのロゴみたいな置き方ができるわけね）とかのデータを加えてこれに通してやるとあーら不思議，プリンタからまるで印刷物のような美しい文書が出てくるってなものなんです。

というわけで，実はついに永年あこがれていたX68000用のTEXを手に入れたんですよ。うるうる，うれすい……。で，うれしさついでにG++コンパイラもネットからダウンロードしてきちゃった。こいつはGNU版のC++モドキで，早い話，GCCの親分みたいなもんなんですけどね。

しかし……，TEXって馬鹿でかいよー。フロッピーディスクにして8枚分（10Mバイト近い)。20Mバイトしかない私のハードディスクはもうどんなに詰めてもあと2,3Mバイトしか余らない。ううっ，どうやってインストールしろっちゅうんじゃ。

しょうがないから，G++で遊ぼうと思ってドキュメントを読むと……，なに，まともにG++を動かすには4MバイトはRAMが必要だと（2Mバイトでもぎりぎり動かないことはないらしいんだけど)。私のマシンはこの前，SX-WINDOWと一緒に2Mバイトになったばっかしなんだぞ。

ということで，そのような環境になるまではおあずけとなったのでした（しかし，いつになることやら)。

つめてつめてるプロモーション

ではでは，今月の1本目。今月の1本目はちょっと便利なX-BASIC用関数，p_symbol( )です。

p_symbol( ) for X68000
(X-BASIC)
富山県　作田定之

プロポーショナルピッチってご存じですか？　そそ，プリンタのコマンド(モード)なんかにありますね。英語で文字を打ち出すときに字と字の間を詰めてかっこよく見せる，あのプロポーショナルピッチです。

このp_symbol( )はそのプロポーショナルピッチで英文字と記号（全部の記号というわけじゃないんだけどね）をプロポーショナルピッチでグラフィック画面に表示する関数なのです。

書式は簡単で，

p_symbol(p1,p2,p3,str,h,v,p4,p5,p6)

となっています。引数にはそれぞれint型（ただし，strだけは文字列型）で，

p1	X座標
p2	Y座標
p3	文字間隔をドット単位で指定
str	表示する文字列
h	横倍率
v	縦倍率
p4	モード　0=16×16　1=24×24
p5	パレット・コード
p6	表示=1 or 非表示=0

という内容を入れてください。

返り値は整数で表示した場合の横幅のドット数をint型で返してきますので，うまくp6を応用すればセンタリングなども簡単にできるんじゃないかと思います。

p_symbol( )のサンプル

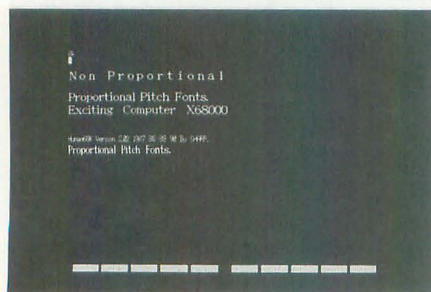
リスト1がp_symbol( )で，リスト2がp_symbol( )を使ったサンプルプログラム。リスト1で打ったものを（たとえば，リスト1を打ってpsymbol.basでsave@したのなら)，

```
load@ "psymbol. bas"
renum 200
```

などとして，200行以降に関数を持ってきてからリスト2を打ち込んでください。

おお，かっこええじゃあないですか。プロポーショナルピッチか，やりますねえ。文字間を詰めるだけでこうもかっこよくなるもんなのね，英文字ってのは。サンプルの実行結果を見てるとほれぼれしちゃいますよ。グラフィック画面にLINEやCIRCLEでよいしょとお絵描きして，こいつで文字を画面に表示し，プリンタに打ち出せば……，いま流行りのデスクトップパブリシングだ（おおっー）！

それは冗談としても，作者の作田さんががんばりましたね。このプログラム，英文字の字間を詰めていくわけですが，詰めるためにはやっぱり文字の大きさをプログラムが知ってなきゃいけないわけですよね。てことは作田さん，文字の大きさを決めるために1つひとつ試行錯誤して決めていったんでしょうか（それともなにか別の方法？)。もし，そうだったらすごく手間ひまかかったんじゃないかと思います。その努力に金メダルあげましょう。めざせTEXへの道だあっ！

おっと，そうだ。この関数はあくまでSYMBOL関数，グラフィック画面に描画する関数ですからね，忘れてグラフィックRAMにRAMディスクなんか設定しちゃいけませんよ。それを忘れた私は……，せっかくRAMディスクにセーブした原稿をみーんなオシャカにしちゃいました。これ，実はいまあわてて書き直してる最中だったりして。うおーっ，私の原稿を返せーっ！　気をつけましょうね。ぐっすし（そんなや

つはお前だけだって)。

LIVE in ショートプロぱーてぃ

さーて，気を取り直して次いきましょうか。今月の2本目はこのページでは初登場のX68000用のバッチファイルによる作品，「play.bat」です。

play.bat　for　X68000
(Human68k)
島根県　東裕人

OPMファイル(ファイルの拡張子の3文字が＊.opmとなっている音楽データファイルのあれのことですね)を演奏するためのバッチファイルです。

このバッチファイルの実行には「play.bat」本体とデータの「init.opm」がペアで必要になります。で，「init.opm」はなく，それを作成するBASICプログラム「makeinit.bas」(リスト4)を実行する必要があります。

1) ed.xを立ち上げる
2) リスト3「play.bat」を打ち込む
3) [ESC]，[T]でファイル名変更になるので“play.bat”と打つ
4) ディスク上にplay.batができる
5) BASICを立ち上げる
6) リスト4「makeinit.bas」を打ち込む
7) RUN

という手順で，間違いがなければディスク上にinit.opmが出来上がるはずです。

おっと，注意！「makeinit.bas」はシステムを立ち上げてまだなにも曲を鳴らしていない状態(内蔵音色が壊されていない状態)で，実行してくださいね！

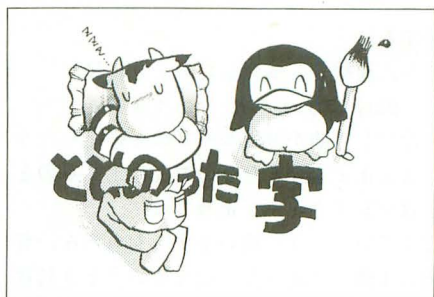
このように準備にはちょっと手間取るかもしれないけど，使い方は結構簡単です。コマンドモードで，

A>play [OPMファイル名/スイッチ]

とすればOKです。これで指定したOPMファイルを演奏してくれます。ファイルを指定しない場合は，現在のトラックにあるデータで演奏するようになっています。

で，肝心のスイッチなのですが，以下のようなものがあります。

/i　　初期化。OPMドライバの内蔵音色，およびノイズジェネレータの初期化を行います

/n　　ノイズジェネレータの初期化。ノイズジェネレータの初期化のみの動作になっています。曲も止まってしまうので，注意して使うこと

/s　　演奏の停止。ただし，停止するだけでトラックの内容は壊しませんからplay(cr)や play /c などで，また演奏を開始することができます

/c　　演奏を停止したところから再開

/o nn 演奏中の曲のテンポの変更(ただし，ファイルの中にテンポの指定があるとまた元に戻ってしまう)。nnは32から200の間で指定してください。また，oとnnの間にはスペースが必要です

/?　　ヘルプ

実際に使うには，

play /i ファイル名

(音色を戻しておいてから，ファイルを演

リスト1

```
 /*      プローポーショナル・ピッチ対応 symbol 関数
 /*
 /*      書式:
 /*             p_symbol( p1 , p2 , p3 , str$ , h , v , p4 , p5
 , p6 )
 /*             戻り値 = 横幅
 /*
 func p_symbol( x , y , z , st$;str , h , v , mo , plcode , sw )
  int i,j,k,l,p,s
  int xpt
  int X
  str m$
  dim int lp( 1 , 104+20+21*2 ) = {
   +0,
   +1,2,3,2,2,2,2,2,5,4,1,2,0,1,2,2,2,2,3,2,1,1,0,2,1,2, /* 16x16
   +4,4,4,3,4,4,3,4,6,5,3,6,1,3,4,4,3,5,4,5,3,3,1,3,3,3,
   +3,5,3,3,2,3,3,3,3,3,
   +6,1,1,3,0,3,1,3,2,0,2,0,6,6,1,2,2,3,1,5,2,
   +0,0,0,0,1,1,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, /*  8x16
   +0,1,1,0,0,1,0,1,2,1,1,2,0,1,0,1,0,2,1,1,0,0,0,0,1,1,
   +1,1,0,0,0,0,0,0,0,0,
   +3,1,0,0,0,0,2,1,0,2,0,2,2,2,1,0,1,0,0,0,1,
   +0,
   +2,4,3,3,3,4,3,3,8,5,3,4,2,3,3,4,3,3,5,2,3,2,0,3,2,3, /* 24x24
   +5,4,5,5,5,6,5,4,8,5,4,9,2,5,5,4,5,6,6,6,4,4,1,4,4,5,
   +5,7,5,5,3,6,5,5,4,5,
   +10,2,3,5,1,2,4,5,3,4,3,4,10,10,2,3,2,6,3,9,17,
   +0,1,1,1,1,1,1,0,3,1,1,1,0,1,1,1,1,0,1,1,0,0,0,0,1,1, /* 12x24
   +0,0,1,0,1,1,0,0,4,2,0,4,0,0,1,0,0,1,1,1,0,0,0,1,0,1,
   +1,3,1,1,1,1,1,1,1,1,
   +4,1,1,1,1,0,3,1,1,2,2,2,4,3,2,1,2,1,1,1,4 }
  /*
  dim int rp( 1 , 104+20+21*2 ) = {
   +0,
   +2,4,3,3,3,3,3,2,6,4,3,3,1,2,3,4,2,3,4,3,2,2,1,3,2,4,
   +5,4,5,5,5,6,4,4,7,7,4,7,2,4,5,4,5,4,5,5,4,4,2,4,4,5,
   +4,6,4,4,4,4,4,4,4,4,
   +7,10,2,4,1,2,13,4,3,14,3,14,8,8,3,3,2,4,1,6,2,
   +1,1,1,1,1,1,1,1,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
   +1,1,1,2,1,1,1,1,3,3,1,4,1,1,1,1,2,1,1,2,2,1,1,1,1,1,
   +1,2,1,1,1,1,1,1,1,1,
   +4,2,1,1,1,1,4,2,1,4,1,4,4,4,1,1,1,1,1,1,4,
   +0,
   +2,4,4,4,4,4,2,3,8,3,3,3,2,3,3,4,2,2,4,2,3,2,0,2,2,4,
   +4,5,6,4,6,6,4,4,8,9,4,9,2,4,5,5,4,5,6,6,5,4,1,3,4,5,
   +5,7,5,5,5,5,5,5,4,5,
   +11,14,3,4,1,2,17,6,3,17,3,17,11,11,2,3,2,6,3,9,4,
   +1,1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,
   +1,1,2,1,1,2,1,1,4,3,1,4,1,1,1,1,1,2,1,2,1,1,1,1,1,1,
   +1,3,1,1,1,1,1,1,1,1,
   +4,2,1,1,1,1,5,1,1,6,2,7,5,5,2,1,2,1,1,1,2 }
  /*
  str chkstr$[256]= "A B C D E F G H I J K L M N O P Q R S T U V
 W X Y Z"
  chkstr$=chkstr$ + "a b c d e f g h i j k l m n o p q r s t u v
 w x y z"
  chkstr$=chkstr$ + "0 1 2 3 4 5 6 7 8 9"
  chkstr$=chkstr$ + "! " # $ % & ' * + , - . : ; < = > ? @ ^ ‘"
  chkstr$=chkstr$ + "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqr
 stuvwxyz"
  chkstr$=chkstr$ + "0123456789"
  chkstr$=chkstr$ + "!"
  chkstr$=chkstr$ + chr$(&H22)
  chkstr$=chkstr$ + "#$%&'*+"
  chkstr$=chkstr$ + chr$(&H2C)
  chkstr$=chkstr$ + "-.:;<=>?@^`"
  /*
  if mo=0 then xpt=16
  if mo=1 then xpt=24
  X=x
  for i=1 to len(st$)
   k=asc(mid$(st$,i,1))
   if (k>&H1F and k<&H80) or (k>&H9F and k<&HE0) then j=1:s=167:l
 =83 else j=2:s=1:l=-1
   m$=mid$(st$,i,j)
   p=instr(s,chkstr$,m$)
   p=(p¥j-l)*(p¥p)
   if sw=1 then symbol( x+z*h-(lp(mo,p))*h , y , m$ , h , v , mo+
 1 , plcode , 0 )
   x=x+((xpt¥(3-j))+z-(rp(mo,p)+lp(mo,p)))*h
   i=i+(j¥2)
  next
  return(x-X)
 endfunc
```

リスト2

```
   10 /* sample
   20 /*
   30 screen 2,0,1,1
   40 /*
   50 str a$[80]="N o n   P r o p o r t i o n a l
   60 str b$[80]="P r o p o r t i o n a l  P i t c h  F o n t s

   70 str c$[80]="E x c i t i n g   C o m p u t e r   X 6 8 0 0
 0
   80 str d$[80]="Human68K Version 2.02 1987 88 89 90 By SHARP.
   90 str e$[80]="Proportional Pitch Fonts.
  100 /*
  110 palet(1,rgb(31,31,31))
  120 symbol(0,50,a$,1,1,2,1,0)
  130 p_symbol(0,100,1,b$,1,1,1,1,1)
  140 p_symbol(0,130,1,c$,1,1,1,1,1)
  150 p_symbol(0,200,1,d$,1,1,0,1,1)
  160 p_symbol(0,220,1,e$,1,1,1,1,1)
  170 end
  180 /*
  190 /*
```

奏する)
とか，連続して演奏したいときには，

play ファイル名 ファイル名……

(ただし，曲が無限ループのものだったりすると止まらなくなるので注意！ そのときはブレイクキーで止めてね)
なんていうふうに使います。もちろん，普通に１曲だけ聞きたいなんていうときにはシンプルに，

play ファイル名

で，OKです。そうそうHuman68k ver.1.0ではバッチのためのメモリが足りなくなり，バッチエリアサイズを拡大(COMMAND/B：8などと)しなければ動かないので注意してくださいね。

おお，そうだ。このプログラムでは，

play ファイル名

とすると，OPMファイルをカレントディレクトリに探しにいきますが，そこに指定したOPMファイルがないときには，“music”という環境変数に入っているディレクトリから探してきます。環境変数がどんなものかはマニュアルを見てもらうとして，とりあえず，たとえばOPMファイルを入れているディレクトリが“opmfiles”というディレクトリ名であれば，

set music = a:¥opmfiles

というふうに設定することができます。“autoexec.bat”にでも入れとくと便利でしょう。

なになに？ 投稿原稿によれば，「Oh! X10月号113ページを見てこれならもっと便利なバッチファイルがうちにあるぞい。と思い，投稿することにしました。内容からいっても，これは「ショートプロぱーてぃ」宛に投稿するのが妥当だと思いますので，古村さん，煮るなり焼くなりしてください」とのこと。じゃ，私はライターであるから焼いてみようか？(こらこら)

さて，この「play.bat」はこのページ初のバッチファイルになるわけです。私はどんな言語でも受け付ける！ と言ったし，サ

リスト3

```
echo off
break kill
set $plf$=FALSE

if "%1" == "" goto PLAYN
goto SKIP

:LOOP
if "%2" == "" goto END
shift
:SKIP
if "%1" == "/p" goto PLAYN
if "%1" == "/P" goto PLAYN
if "%1" == "/s" goto STOP
if "%1" == "/S" goto STOP
if "%1" == "/c" goto CONT
if "%1" == "/C" goto CONT
if "%1" == "/?" goto HELP
if "%1" == "/n" goto NOISE
if "%1" == "/N" goto NOISE
if "%1" == "/o" goto TEMPO
if "%1" == "/O" goto TEMPO

if %$plf$% == FALSE goto PLAY
echo (w) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul

:PLAY
if "%1" == "/i" goto INIT
if "%1" == "/I" goto INIT
if exist %1 goto PLAY1
if exist %1.opm goto PLAY2
if exist %music%%1 goto PLAY3
if exist %music%%1.opm goto PLAY4
echo << ファイルがありません >>
goto LOOP

:PLAY1
echo << %1 を演奏します >>
copy %1 opm > nul
goto PLAYE
:PLAY2
echo << %1.opm を演奏します >>
copy %1.opm opm > nul
goto PLAYE
:PLAY3
echo << %1 を演奏します >>
copy %music%%1 opm > nul
goto PLAYE
:PLAY4
echo << %1.opm を演奏します >>
copy %music%%1.opm opm > nul
:PLAYE
set $plf$=TRUE
goto LOOP

:PLAYN
echo << 演奏を開始します >>
echo (p) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul
set $plf$=TRUE
goto LOOP

:STOP
echo << 演奏を停止します >>
echo (s) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul
set $plf$=FALSE
goto LOOP

:CONT
echo << 演奏を再開します >>
echo (c) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul
set $plf$=TRUE
goto LOOP

:TEMPO
echo << テンポを変更します >>
echo (o%2) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul
set $plf$=TRUE
shift
goto LOOP

:INIT
copy c:¥batch¥init.opm opm > nul
echo << ＯＰＭドライバを初期化しました >>
set $plf$=FALSE
goto LOOP

:NOISE
echo (i)(m1,10)(t1)y15,0(a1,1)(p)(i) > %temp%$opmcmd.$$$
copy %temp%$opmcmd.$$$ opm > nul
del %temp%$opmcmd.$$$ > nul
echo << ノイズジェネレータを初期化しました >>
set $plf$=FALSE
goto LOOP

:HELP
echo Play ver 4.01
echo 使用法：play [ファイル名／スイッチ]
echo    /i      ＯＰＭドライバの初期化
echo    /n      ノイズジェネレーターの初期化
echo    /p      現在のトラックの内容を演奏する
echo    /s      演奏を停止する
echo    /c      演奏を再開する
echo    /o nn   テンポを変更する(32<nn<200)
echo    何も指定しない時は現在のバッファの内容を演奏する

:END
set $plf$=
break on
```

リスト4

```
 10 /* make "init.opm"
 20 /*              for play.bat
 30 /*
 40 str filename = "init.opm"
 50 int vmax = 68
 60 int c1, c2, c3, fp
 70 str crlf, inidata[255]
 80 dim char vdata(4,10)
 90 crlf = chr$(&HD)+chr$(&HA)
100 inidata = "(i)"+crlf+"(m1,10)"+crlf+"(t1)y15,0"+crlf+"(a1,
1)"+crlf+"(p)"+crlf
110 /*
120 fp = fopen(filename, "c")
130 fwrites(inidata, fp)
140 for c1 = 1 to vmax
150   m_vget(c1,vdata)
160   fwrites("(v"+str$(c1)+",0,", fp)
170   for c2 = 0 to 4
180     for c3 = 0 to 10
190       fwrites(str$(vdata(c2,c3))+",", fp)
200     next
210   next
220   fseek(fp,-1,1)
230   fwrites(")"+chr$(&HD)+chr$(&HA), fp)
240 next
250 fclose(fp)
260 end
```

イズも100行ちょっとの大きさであります から，もちろんショートプロの範囲内。ばっちりOKのプログラムであります。

UNIXなんかではこういうバッチファイルみたいなものがシェルプログラミングとかいって結構盛んなジャンルなんですよね。たとえば，Cシェルなんてのはその名のとおりCもビックリというようなすごいシェルスクリプトの文法になっているんです。だから，ガンガンこういうのも送ってきてくださいね。

いま思い出したんだけど，先月に続いて今月もX1のプログラムがなかったなあ。みんなX1のプログラムもどんどん送ってきてくれい。そんなこんなでまた来月。

(で)のぱーてぃハンズ第2部——（その3）

前回はえっと，ぱらぱらぱら……，そうか自分の視界の話をしたんでしたね。そういうことです，はい(いきあたりばったり)。とりあえず今月は前回の解説の続きです。プログラムリストも先月号に出てしまってますから，先月号を用意してくださいね。

前回は視界を決めてそのための配列を用意したんですよね。ところがどっこい，視界があるからといってその範囲のものがなんでも見えるわけじゃない。え，なぜかって？　そりゃね，たとえば目の前にかわいい女の子が立ってるとしますでしょ。で，普通は洋服があるから，その先は透けて見えたりしないわけだ。透けて見えちゃったら困るでしょ。でへへ。

……まあ，変な説明ですけど，つまりは手前にものがあるときはその奥のものが見えてしまってはいけないわけです。というわけですので今月はその部分，視界がじゃまものに遮られているかどうかのアルゴリズム，いきます。

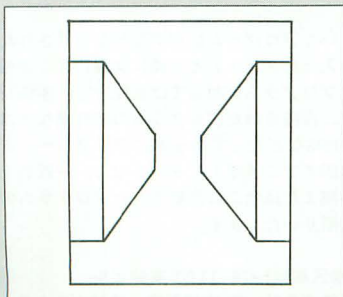
私はぬりかべっ！

ある壁がほかの壁を遮っちゃう，というとそう，中央の壁が目の前にでーんと立ちはだかっているときなんてのがありますね。私はぬりかべ！　つまり中央の配列で手前側に壁があるとその後ろにいくら壁があっても見えないんですね。それは中央だけではなくて左右の配列もそうですね。

配列でいうと(憶えてますよね，先月の配列) 5番に壁があったら2番や11番に壁があっても描く必要がないわけです。うむ簡単簡単。

しかし，後ろにある壁が遮られるのはそれだけじゃあない。たとえば，図1のように見えたとしましょう。

左側の壁でこちらを向いている壁がありますよね。こいつが見えなくなるときがあるでしょ，中央の壁がないときでも。そうです。図2のように「もうひとつ手前の左手に壁があるとき」ですよね。

図1

つまり「奥の壁のこちらを向いている面は，その前の側面に壁があるときは見えない」わけです。右側でも同じ。

「左右の壁では手前側に続いて壁があるかないかで，奥側の壁の手前に向かった面を描くかどうかが変わる」ということになるわけです。配列でいうと1番の配列には壁がなくて，2番にもなければ何も描かなくていいけど，もしあったらその部分を描いてやらなくてはいけないわけです。

アルゴリズムは漬物

この2つのことを考えながらプログラムの流れ，アルゴリズムを作っていきます。えーっと，まず，中央に壁がある場合だから中央のどこに壁があるかを調べるんだよね。

えーっと，まず

1）中央の壁を（配列でいうと4〜7）壁があるところを探す

2）中央の壁を描く（このとき深さがいくつだったかを記憶しておく）

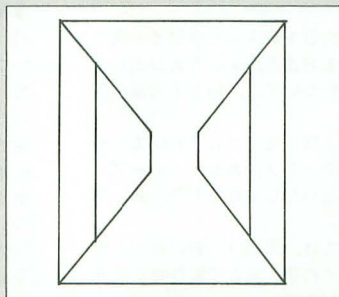
ということをします。

で，左右の壁だよね。まず，左を見ていくことにしますが，さっき中央の壁を見たから，さっき壁のあった1個手前まで（図1と図2を考えたら，配列で7のところに壁があるなら，お隣のところに壁があっても関係ないでしょ。だからその前までしか調べないのだ）調べればいいわけね。

3）さっき壁のあった左隣の配列の1つ手前まで調べる

で，さっきのその前の壁を描くか描かないかを調べる。ま，ここはifで場合分けするのが無難かな？

4）もし，壁があったらそれを表示。壁がなくてその前にあったら，前の壁のこちらに向いている面（縦に長いあれね）を表示する

図2

……当然，前にもなかったらなにも描かない

5）以上を繰り返す

でいいかな？　うむうむ，よさそうだ。じゃ右側の壁は，

6）左に同じ

でいいわけだ。よかったよかった。

アルゴリズムを作るときにしっかり考えて作っておくとプログラムを組むときに楽ですし，逆にいい加減にやっちゃってこの段階でバグがでると致命的ですんで，しっかりこの段階でプログラムを組むときのことまで考えてやりましょうね。ゆっくり時間をかけて。ただし，忘れてしまわない程度に時間をかけて，だけど。

やっとプログラム

という方針で，できたのが先月のリストなわけですね。

ループを組むときにforループではなくてwhileループを組んでいるところがあります。これはなんでかっていうと，forループというものは条件がおかしくても必ず1度は通ってしまうものと思い込んでいたからなのです。そして，自分が「石の中にいる」ときは左右は関係ない＝つまり左右を調べてはいけないということがあるから，forループではだめ，と思ってしまったのです。いやあ，失敗失敗。

おおそうだ。すっかり忘れていた。中央左右について壁の遠さを入れるとその絵を描いてくれるサブルーチンを作りました。せっかく絵のパターンを作ったんだから，エレガントに呼び出そうと思ったわけですね，はい。うーん，にしてはCASE文でだらだら書いちゃったら，なんか長くなっちゃったなあ。うーん，もうちょい検討の余地があったかな？

しかし，ま，なんとかプログラムが形になってきたでしょ。

とりあえず，この時点でやっと配列に0か1か2を入れてrunするといろいろ描いてくれるところまでできました。なんとなくダンジョンぽくなったでしょ。確認のためにもいろいろ配列に数値を入れて遊んでみてくださいね（先月の時点でやっちゃったかな？）。

えっと，来月はダンジョンのマップを20×20の大きさで作って，再来月はおまけの処理をちょっとつけて……。げ，やっぱり6回の予定だったのに1回足りない……。こいつはまいった，どうにかしなきゃだわ。

まあ，とにかくそういうことでまた来月。てやっ！（と頭を抱えて去る）

投稿プログラム大募集
のお知らせ

●あなたはどのようにしてプログラムに出会いましたか？

記号の羅列にすぎなかったプログラムリストが突然意味を持ったメッセージとして読み取れる，それを機に「プログラム」というものについてなにか納得できるようになる……。きっかけは雑誌のページの隅に載った小さな小さなプログラムだったのかもしれません。またはいくら見直してもエラーの出てくる長いBASICプログラムかもしれません。きっとそのプログラムにある「なにか」に魅かれてリストを打ち込んだことがあると思います。

あるソフトを使っていて，なにかの記事を読んでいて，または突然に，「こんなソフトがあったらいいな」と思う。こういった小さな動機からプログラムは生まれてきます。あなたのアイデアを埋もれさせないでください。私たちはそこにある「なにか」を求めています。完成度の高いありふれたプログラムよりも，粗削りでもオリジナリティの光るプログラムのほうが，さらに誰かの「プログラム」を生むことになるはずです。

Oh!Xには毎月さまざまな投稿プログラムが掲載されています。これらのプログラムは，すべて読者の皆さんが日頃のパーソナルコンピューティングのなかで作り上げてきたものです。あなたも投稿プログラムを通じてOh!Xの誌面作りに参加しませんか？

●大作歓迎！

Oh!Xでは過去に40Kバイト程度のプログラムまで誌上に掲載した実績があります。また，どうしても誌面に載り切らない作品は付録ディスクに収録して配布したこともありました。どうせ誌面には掲載できないからと諦めている方，とりあえずご連絡ください。

X68000用　カードゲーム　TEN

X68000用　かべくずし

MZ-700用　Eyelarth

全機種共通システム用　BILLIARDS

X68000用　XROT0. X

X68000用　ハンディイメージスキャナアダプタの製作

投稿募集要項

1)　お送りいただくプログラムには，住所，氏名，年齢，職業，連絡先電話番号，機種名，使用言語，動作に必要な周辺機器，マイコン歴などを明記のうえ，封書の宛先の最後には「Oh!X LIVE」，「全機種共通システム」，「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2)　投稿されるプログラムには詳しい内容を記入した原稿と一緒に変数表，メモリマップ，参考文献などもお書き添えのうえお送りください。また，お送りいただいた原稿については，当方で加筆修正をさせていただくことがあります。

3)　お送りいただくプログラムは最低2回はセーブしておいてください。基本的に同封されたフロッピーディスク，カセットテープ，クイックディスクなどについてはご返送いたしませんので，あらかじめご了承ください。

4)　ハード製作関係の投稿につきましては，最初は内容のわかる原稿のみお送りいただければ結構です。その後，当方で製作物が必要だと判断した場合には改めて連絡いたします。

5)　お送りいただいた作品の採用につきましては，掲載号が決定した時点で当方より連絡いたします。特にツール関係，ハード関係などのものにつきましては特集内容などを考慮したうえで採用決定されますので，結果を連絡するまでにかなり時間がかかる場合があります。

6)　投稿いただいたプログラムにバグなどが発見された場合は新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。

7)　掲載されたプログラムに対しては当社規定の原稿料をお支払いいたします。また，投稿されたプログラムの著作権などは制作者に保留されますが，PDSなどとしてネットにアップロードされる場合は必ず事前に編集部までご連絡ください。なお，一般的モラルとして，他誌との二重投稿または他誌に掲載されたプログラムの移植などについては固くお断りいたします。

宛先
〒108　東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク株式会社　Oh!X編集室「投稿プログラム」係

X68000 マシン語プログラミング 入門編

著●村田敏幸

B5判変型・388ページ　定価2800円(税込み)

ストロングスタイルのマシン語入門!!

●本書まえがきから

1990年11月現在。『Oh！X』誌に連載中の「X68000マシン語プログラミング」のうち＜入門編＞と称した冒頭部分を1冊にまとめたら，こんな本になった。マシン語プログラミングに興味をもったX68000ユーザーのための副読本，とでもいったらよいのだろうか。少なくとも，教科書的なプログラミング入門書では決してない。むしろ，問題集であり，実践テキストのノリに近い。

マシン語にかぎらず，プログラミングに関する知識/技術は，実際のプログラミングの中でこそ身につき，磨かれるものだ。この不変の真理にもとづき，本書は読者に自分の頭と体とを使うことを強いるように書かれている。エッセンスを100倍くらいに薄めて吸い差しでとろとろと流し込むような親切さは排除した。文書の襞に隠れた大小の謎は，サンプルプログラムを読み，動かし，改良することによって解き明かされるだろう。

●本書の内容

'91年1月刊行予定

SX-WINDOW プログラミング

吉沢正敏●著

B5判変型　予価3200円(税込み)

X68000にイベントドリブン方式のマルチタスク・ウィンドウ環境を提供するSX-WINDOWは，X68000に新たな世界を拓くものとして熱い期待を集めている。本書は，このSX-WINDOW上でプログラムを作ってみたいと思っているユーザーを対象にした，プログラム作成のためのガイドブックである。イベントドリブン，リソースなどのウィンドウ・プログラムの基礎知識，サンプルプログラムによる具体例，ウィンドウ関連のシステムコール一覧など，SX-WINDOW上でプログラミングする際のエッセンスを集めている。

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1991年2月18日の到着分までとします。当選者の発表は1991年4月号で行います。

❶ イメージファイト

アイレム ☎06(535)4880

X68000用 5″2HD版2枚組

9,700円(税別) 3名

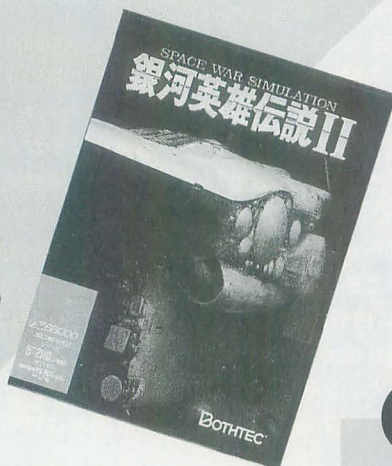
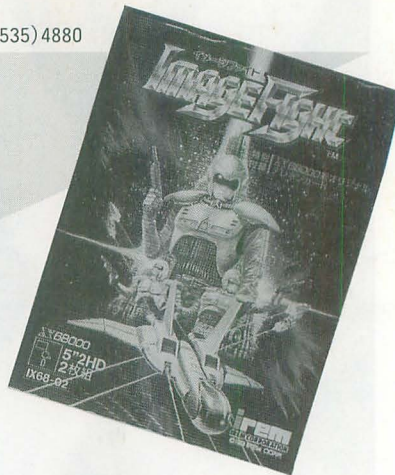
ファン待望の超ムズシューティングゲーム。ゲーセンでお金を注ぎこんだ人もかなりいるはず。もちろん移植の出来は期待どおり。シューティングファン必携の1作だ。

❷ 銀河英雄伝説II

ボーステック ☎03(3708)4711

X68000用
5″2HD版4枚組

9,800円(税別)

3名

ご存じ「銀河英雄伝説」の続編。システムもさらにパワーアップされ，シミュレーションに慣れていない人でも手軽に遊べる。全自動モードやMIDI対応もうれしいかぎりだ。

❸ ポピュラス プロミストランド

X68000用 5″2HD版

4,800円(税別) 3名

すでに持っている人も多いかもしれないけど，このプロミストランドは，ただのプロミストランドではなぁい！ なんとあのピーター・モリニュー氏のサイン入りなのだ。ってことで3名の方に。

❹ 大戦略卓上カレンダー

システムソフト ☎092(752)5262

5名

大戦略シリーズに登場する戦車や戦闘機のPHOTOがプリントされた，システムソフトオリジナルの1991年度カレンダー。これを見るたびに大戦略をプレイしたくなる？

❺ イースのテレホンカード

日本ファルコム ☎0425(27)6501

3名

お馴染みイースのキャラクターテレホンカード。大事にとっておくもよし，見せびらかしながら使うもよし，定期入れにしのばせてプロマイド代わりにするもよしのテレカだ。

12月号プレゼント当選者

1シースルークロック（新潟県）金原真也（福岡県）升井晋也 2クリスタルポルシェ（長野県）宮尾勇（新潟県）前田育男 3ジッポ・ライター（福島県）佐藤明広（千葉県）正木崇穏（神奈川県）多田早利 遠藤幹文（奈良県）山下日出夫 4キーホルダー（東京都）鈴木善統（神奈川県）鈴木康之（大阪府）土器彰信 友広一郎 伊藤一成 5ネクタイピン（岩手県）今井純二（静岡県）玉宮央善（大阪府）二宮善弘（香川県）氏家啓（福岡県）古川智雄 6ポーチ（北海道）鈴木賢吾（長野県）中沢純 7ボストンバッグ（東京都）内田大輔（愛知県）高橋大介 8XBAStoC CHECKER（京都府）岩成英一 9C compiler PRO-68K（広島県）森崎剛 10CANVAS PRO-68K（東京都）宮島誠 11熱血高校ドッジボール部サッカー編（埼玉県）山賀厳太郎（静岡県）遠藤慎弥（沖縄県）大城久

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

（価格はすべて消費税別です）

[第9話]

街の空気

TAKAHARA HIDEKI 高原 秀己

この年末年始に海外に出かけた人は，昨年よりも10%ほど増えて45万人以上に及んだそうだ。昨年は海外旅行に出かけた人が初めて1年間に1000万人を超えたとか。海外旅行ブームの拡大を象徴している。

特にこの年末年始の場合，休みが散らばっていたのでうまく有給休暇や代休を組み合わせたりすれば，対象期間が12月22日から1月6日までと実に16日間に及ぶ，従来にないロングな冬休みになった。もちろん16日続けて休めるなんて人はそうそういないが，昨年末か今年始めのどちらか一方は続けて休みが取れる，ということはある。

年末年始は狂ったように同じ期間にみんなが旅行に出かけるものだが，今回はそのせいか，やや集中回避。12月22日にオーストラリア1週間の旅に出かけて新年は自宅だとか，年が明けてすぐに成田に向かい，香港4日間の旅行に出かけたとか，従来とは違ったパターンの海外旅行を楽しんだ人もけっこういたそうだ。

なにしろ政府をあげての長期休暇奨励という奇妙なご時勢。いままでほど必死になって働いてはいけないそうなのである。たとえば「丸の内の不夜城」といわれる，コンピュータ業界で有名な某社。仕事は3年以上先の分までたまっているらしいのだが，世間体があるのを気にしてか，深夜残業や休日出勤は極力しないように，との通達が出されている。

この会社に限らず全般的な話だが，無趣味で出不精な人が突然，長期休暇などを手にしては，逆に困るのである。することがまずない。

そんな人にとっても問題なく効果的に時間を使えるのが旅行だ，という解説を聞いたことがある。確かにそのとおりだろう。

もっとも日本人は元来，旅行好きだという説もある。とにかく映画，テレビ放送，出版を問わず，トラベルものなら当たりやすいといわれているのは常識。昨年30年間続いて幕を閉じ話題となった「兼高かおる世界の旅」もそうだが，「新世界紀行」にしても「世界ふしぎ発見」にしても，この種の番組の人気の高さがわかる。

もうひとつの人気ネタ，ミステリーとドッキングすると，ヒットする確率はさらに高まる。出版界では「トラベルミステリー」などと呼ばれており，西村京太郎氏から内田康夫氏まで作家側もはっきりと意識して作品を量産している。

こうした作品が2時間ドラマになると，「中国殺人行脚　萩～広島～姫路，花の中年奥様3人組が次々と遭遇する女子大生温泉連続殺人事件　旅先でまき起こる恐怖の……」といったやたら長い題名のドラマに化ける。バカバカしい感じはするのだが，知らず知らずのうちに見終わっていることがあるから，人のことは笑えない。

お正月映画として今年も「男はつらいよ」シリーズが上映されているが，トレンディな洋画に劣らず人気があるのは，寅さんもやはりトラベルものの典型だからか。

◇

そのお正月映画。

「ディック・トレイシー」，「トータル・リコール」，「ロッキー5」，「ネバーエンディングストーリー第2章」となかなか豪華な布陣が揃っている。「ネバーエンディングストーリー第2章」を除けば，昨年夏の米国興行街を賑わせた作品がズラリと揃っている。ここに「バック・トゥ・ザ・フューチャーPart 3」や「アラクノフォビア」，「ゴースト」が加わっていたのだから，昨年夏の米国映画市場が大混戦のデッドヒートになったのは無理ないところ。

レースを制した伏兵中の伏兵「ゴースト」は我が国ではひと足早く秋に公開ずみ。確かに快作だったが，米国にせよ日本にせよ，よくあれだけ動員できたものだといまだに感心する。

昨年夏の米国公開映画の中で最も期待していたのが，ウォーレン・ビーティ監督兼主演の「ディック・トレイシー」。「ゴースト」に食われはしたものの，あれだけの強豪揃いの時期に上位をキープしていた（最終的には「トータル・リコール」に次ぎ3位)のだから，ヒットしたことは間違いない。

で，見た感想なのだが，それほどではなかった。というか期待外れだった。

見終わったあと，「バットマン」と類似した感想を抱いてしまった。双方とも舞台となっているのは架空の暗黒街で，街を支配する悪党一味と戦う正義の味方，という物語。禁酒法時代の頃であろうか，おそらく同じ頃の米国暗黒街をモデルにしているようだ。

この街の模様が両作とも実に似ていた。おまけに主人公が活躍するのがほとんど夜で，ライティングまで同じタッチときては感想が似てくるのはしかたないかも。

とにかくこの街の舞台作りや演出がしっくりこず，印象をかなり左右された。この種の米国人の論評は見たことがないためなんともいえないが，もしも米国人には自然な感じだとすれば，「街の空気」に対する文化的な日米の違いがあるのだろう。

街の空気，つまり街並みから雰囲気，さらに生活する人々の息づかいまで含めてそう書いたが，この違いは重要だ。今まで一度も行ったことがないはずの場所に親しみを感じる場合，まず似た空気の場所に行ったことがあるはずだという。日本人が日本中，どこに行ってもさほど違和感を感じないのはそのせいで，逆に欧州の人が住む国を変えてもさほど気にしないのはそのせいだろうか。

ここに時代考証が加わってくると，もうどうしようもない。時代劇は論外にしてもたとえば日本の大正浪漫の街並みを米国人に見せてもしっくりこないはず。ちょうど「ディック・トレイシー」や「バットマン」はその逆だったのだろうか。

もっともこうした理由で映画の印象を左右されるようでは，本来はまずい。そう思って「ディック・トレイシー」を回想するのだが，マドンナが妙にハマっていた以外は，どうも印象が弱い。ということは……。

「ディック・トレイシー」以外のお正月映画はまだ1本も見ていない。忙しくて試写状を手に入れる工作をすっかり忘れていたことが，ここにきてたたっている。

ぼくの場合，常設映画評コーナーを持っているわけではないので請求しないと映画会社は試写状をくれないのだ。

第45回　知能機械概論――お茶目な計算機たち――

感涙もののマシン語プログラム

マシン語に興味をもつ少数派

パソコンが一家に1台，さらにひとりに1台というまでに普及してくるにつれて，パソコンの動作原理そのものに興味を持つ人は全体に対する割合としては徐々に少なくなってくると思われます。「パソコンのアーキテクチャなどはどうでもいい，それをどのように使うかが問題である」という思想こそがパソコンをもっともっと多くの人に普及させるためには必要なのですから，パコソンのメカそのものに興味をもつ人が多くならないのはしかたのないことかもしれません。

そのような思想を全世界に普及させようとしている旗手，その人がスティーブ＝ジョブスであります。彼については先月紹介しました。でも逆に，そういう思想が普及してくる時代になればなるほど，アーキテクチャやマシン語など，ブラックボックスをブラックボックスとしてはいられないような姿勢の重要性が増してくると思われます。

今回の記事は，特に本誌の'90年7月の特集「マシン語への第一歩」をがむしゃらに読んだようなタイプの人たちのために書きたいと思います。

コンパイラ自動作成の夢

ソフトウェア危機ということが大声で叫ばれるようになってもうずいぶんと時間が経過しているわけですが，一向にその危機はきていないようにも思われます。というのは間違いで，実際には（特に日本では）ソフトウェアの技術者が膨大に不足しているのです。

新しいマイクロプロセッサチップを完成させるまでの期間はずいぶんと短くなってきましたが，よいコンパイラはなかなか出来上がりません。コンパイラの研究で自動的にコンパイラを作成するというアプローチはひと頃はずいぶんと流行りました。要するに，対象とする言語の記述（どういう構文を持ち，それぞれの構文でどういう処理をし，データタイプは何があるなどということを表現したデータ），それから実行するプロセッサの記述（どういう命令を持ち，どういう処理をし，レジスタは何ビットのものが何本あるかなどというデータ）を入力すると，自動的にお望みのコンパイラが出来上がってしまうというすばらしい話です。

もし，そういうコンパイラ作成ツール（コンパイラジェネレータ）が一度できてしまえば，もうコンパイラのことでやることはなくなってしまうはずでした。新しい言語やプロセッサができても，それを記述するデータを作成するだけで，インスタントに新しいコンパイラが出来上がるのですから。

ところが，実際のところそういう万能ツールはこの世には存在しません。ただし，実際問題としては，本当に作ることができないというわけではなく，手で書いた優れたコンパイラほど，よいコードを吐き出すよいコンパイラが自動的には作れないというほうが正しいかもしれません。このことは，新しい言語やプロセッサアーキテクチャを完全に記述することが困難であるということともかかわってくる問題といえましょう。

パターソン教授の責任

コンパイラジェネレータ実現へのアプローチは決して軽んじてはいけませんが，夢のような話にいつまでも振り回されていてはいけません。ソフトウェア危機を解決する方法はただひとつ，ハードウェアがソフトウェアのことをもっと考えること，と思われてなりません。いくら，ただ速いハードウェアを作ったって，結局は虚しいことになるというわけです。

68000，68010，68020……，あるいは8086，80186，80286……などというマイクロプロセッサのシリーズにおいて，新しいチップが登場するたびに，新機能一覧の中にはいつも「高級言語のサポート」という項目が入っていました。ところが，実際に現場でプログラミングしている人たちは，「新しいチップが登場するたびに逆にプログラミングがだんだんと難しくなってきた」と僕にこぼしたものでした。

命令やレジスタやアドレッシングモードが増えれば増えるほど，少なくとも，選択肢が増えていくわけで，どれを使うかということの重要性は増す一方です。また，その選択自体の仕事も確かに増えるわけですから，そうなったのかもしれません。最適なものをうまく選べば，速度も増し，マシン語もコンパクトになるのでしょうが。

ところが最近のマイクロプロセッサは一体どうなっているのでしょうか？　目を覆いたくなるような現実があるのです。コンパイラなどお構いなしというRISCタイプのマイクロプロセッサが一世を風靡しているのですから。RISC提唱者（首謀者）のパターソン教授本人が，別にコンパイラは誰かひとりが書けばいいのであって，皆さんにご迷惑はかけませんよ，マシン語など書けなくても，いいコンパイラでどんな低レベルの（ハードウェアに密着した）処理でも書けるのですと言って，ソフトウェア危機に対して堂々とシラを切ったわけです。

ところが，どうでしょうか？　うちの研究室などSPARCチップのマシン語に苦しんでいる学生が続出しています（annulビットがどうのこうの，これはdelay slotで……）。「パターソン教授，どう責任をとってくれるのですか？」と言いたいところですが言いません。一般に，大学の研究者など極論すれば，世の中がどうなるかより，自分の論文の数を増やすほうが大事という面が少なからずあるし，それもしかたがないという面もあるようにも感じますので。

RISCチップのワークステーションにUNIXを載せ，Cコンパイラを動かすだけでこの先もよいというならば，文句は言わないのですが……。

乗算プログラムに潜むもの

話がクネクネ化したように思われるかもしれませんが，とりあえず言いたいことは，「さしあたってのところ，マシン語はやはり捨てがたいものがある」という簡単なことなのであります。

ここで，話がいきなり具体的なことにな

り，ひとつの小さなマシン語のプログラム（8ビットと8ビットの乗算）を紹介します。なんでもないプログラムといえばそうなのですが，プログラムを作った人の気持ちが伝わる人には伝わるでしょう。僕が行っている大学の2年生用の実験の中で使用しているプログラムですが，僕自身が作ったわけではありません。

ふつう，マシン語で乗算プログラムを書かなくてはならなくなったとき，まずしなければならないのが，マシン語命令一覧表を調べることでしょう。そして乗算命令がなかった場合には，乗算をいくつかの命令で表現することになります。X掛けるYをマシン語の加算などを使って実現しようとするとき，まずだれでも考えるのが，XをY回足せばよいだろうということでしょう。実際，そのようなプログラムは，冒頭に紹介した本誌の特集の中の毛内さんの記事にも載っています。

ただし，XをY回足すプログラムは大変わかりやすいという特長がある反面，実行スピードという点で満足のいかないものになっています。なぜならば，運が悪いとき，たとえばYが255だったりすると，ループを255回も回らなくてはならなくなるからです。そう考えると，XとYを比較して，XがYより小さいのならば，XとYを入れ替えてYをX回足すという方法がまず思い浮かびます。でもここでは，もっと一気に速度を向上できる画期的なプログラムを紹介することにしましょう。プログラムをリスト1に示します。

このプログラムの意味をここで詳しく説明することはしません。苦労してなんとか意味がわかったときのうれしさを味わってほしいからです。また，マシン語に強い人には説明は不要でしょう。でも，命令の意味だけは書いておかないとだめだと思います。

このプログラムは8085（8080を少しだけ拡張したもの）というマイクロプロセッサ用のアセンブリ言語で書かれています。以下に簡単に命令の意味を書きます。

＊　＊　＊

MVI　B,00H
　Bレジスタの内容を0にする
MOV　C,A
　Aレジスタの内容をCレジスタにコピー
LDA　(KAZU1)
　メモリのKAZU1番地の内容をAレジスタにコピー
DAD　H
　HLの内容（H，L2つのレジスタを連結して16ビットとみなす）とHLの内容の加算（つまり，HLを2倍すること＝1ビット左シフト）
DAD　B
　HLの内容とBCの内容を加算
JNZ
　ゼロフラグがゼロなら指定番地にジャンプ
JNC
　キャリフラグがゼロなら指定番地にジャンプ
SHLD　(KOTAE)
　KOTAEとKOTAE+1で示される番地にHLの内容を書き込む

＊　＊　＊

このプログラムのミソは8行目のDAD Hということができるでしょう。計算の原理そのものは筆算で乗算をするときと類似していますが，筆算とは違って大きいビットから調べているので，この命令ひとつで2つの動作（乗数のビットの切り出しと途中結果のシフト）がなされているのです。

さて，このプログラム中のループを1回実行するときに実行される命令数は分岐命令も含めて4あるいは5という少なさであり，しかもループの回数は乗数がいくつでも（255でも）8回でいいという高速性を誇っています。ループ1回で平均4.5命令実行するとして36という命令数を実行するだけなのです。ちなみに，Oh！Xに載っていたやり方では，仮想CPUのプログラムでは，乗数が128とすると，ループを構成した実行命令数は384命令になります。

実際に速度を厳密に考えるには，単に実行された命令数を数えるのではなく，マイクロプロセッサのデータシートで各命令の実行時間（ステップ数）を調べ，求める必要がありますが，そこまで凝るのは少し大変なことになります。

しかし，このプログラムがベストなものということはできません。まだまだ最適化することができます。たとえば，乗数が小さいときにも必ず8回ループを実行することが気になります。ほかにも改良すべき点はあると思われますが，どんな乗数，被乗数を持ってきても，いつも最小の時間でできるというプログラムを構成するのは至難の技ということができるでしょう。平均（期待値）を小さくするような改良ならば，考えられるでしょうが。

涙の意味

人の手でなければできないような最適化はまだ案外と残っているものです。そして最適化されたプログラムを見るたびに，僕は心の中で涙を流すのであります。それは，「よくもまあ苦労して1命令少なくしたなあ」というプログラマへの賞賛だけでなく，実は，「いずれこのような最適化の技術も定式化されコンパイラに組み込まれるのだろうなあ。結局このような努力は無駄なものとされる日もくるのだろうな」という涙のちょちょぎれるような複雑な気持ちなのです。ですから，「さしあたってのところ，マシン語はやはり捨てがたいものがある」とあえて書いたのです。

参考文献

毛内俊行，「マシン語ってなあに？」，Oh！X 1990年7月号，pp. 47-51

リスト1　乗算プログラム

```
1             MVI    B,00H
2             MVI    L,00H
3             LDA    (KAZU1)
4             MOV    C,A
5             LDA    (KAZU2)
6             MOV    H,A
7             MVI    A,08H
8   LABEL1    DAD    H
9             JNC    LABEL2
10            DAD    B
11  LABEL2    DCR    A
12            JNZ    LABEL1
13            SHLD   (KOTAE)
```

（メモリのKAZU1番地とKAZU2番地の内容の乗算結果をメモリのKOTAE番地に格納）

第56回

猫とコンピュータ
楽しめるRPGギフト

Takazawa Kyoko
高沢 恭子

お世話になったあの方に，という感謝の気持ちをこめて贈るお歳暮。しかし，いまやそれもパソ通で注文する時代らしい。少々味気ないが，忙しい年の瀬には最適か。と思いきや，そうは問屋が卸さないのだ。

厳しさのない，やさしい冬の日々がすぎてゆく。そのぶん，日差しはぼんやりとくすんでいる。季節がその季節らしさを見せてくれないのは気がかりだが，わが子が受験する冬は，おだやかなほうがありがたいなんて思う。

それでも，気候の不順とは別に，東京が年々あたたまっているという話を聞くと，人並みに追想することもある。自分が中学生のころの，あの庭一面の菊の根もとで，黒々とした土を押し上げていたガラスのような霜柱や，コタツに入っていても背中まで押し寄せてきた冷気は，どこへいってしまったのだろうと。

受験の季節がすこし冷えびえとしているのも，あとにおとずれる喜びと旅立ちの春をいっそう輝かしいものにしてくれるためかもしれない。それなら，冷たさに澄み渡る晴天の日も，やっぱりきてほしい。

電気のタタリ

使用4年で火花を吹いて黒コゲになってしまったCRT，PC-KD851のことをFBI-NETにアップロードしたら，テクニカルライターの「ぶん」さんが，こんなふうに教えてくれた。

「たぶんそれは，CRTのフィードバックトランスが，磁束もれを起こしているものと思います。そして，シャーシとトランスの間で放電しているため，バチバチという音が出ているのでは……。はっきり言ってトランス交換しかありません。4年とはかなり短い気がします。よくあるのは，家庭用のTVで，7，8年使用しているものに，この傾向があります。最近は部品代をけちっているので，寿命が短くなっているのでしょうか」

「バイク，くるま」ボードのシグオペ，「Impulse」氏の感想は，「どんな感じだったか想像するだけでちょっと怖いですね。私のはKD-852ですが，PC-8801を使っているときからですから，もう5年ぐらいになるかのかな？ ここ10日くらいはPCの電源入れっぱなしで，CRTもつけたままってこともありますから，いないときにパチパチやられたらこまるなぁ」

「昔，X1turboのディスプレイが，ときどき，『ぱちん！』とかいうことはありました。ほんの一瞬で，CRT 中央に横に1本線が走って……また元にもどります。怖かったけど，ほかに影響がなかったんでほっときました。オレンジ色の火を吹くコンデンサの話も聞いたことがある」というのは「ちゃがま」氏。

「SHUN」君は，「うちのPC-100のCRTも，電源入れたら火を吹いたんだな……修理に出したんだが『修理不能』と言われて，結局，もうカラーCRTが手に入らないから，モノクロでがまんしているのだった」。「Momi」氏も，「うちのテレビは10年近い。うーむ，これはあぶないかなぁ。さすがにヴォリュームがガリオームになってるけど」。

当の黒コゲCRTは，すでに「粗大ゴミ」として区役所に引き取りの依頼をすませてあった。「ぶん」さんは「放電を助けているホコリをとりのぞけば，だましだましもう少しは使える」と言うけれど，いまにも爆発しそうなCRTと向き合ってキーを叩く勇気はおこらない。それに，じつは背中をはがして中を調べたり，通電させてみたりさんざん遊んだあげく，ネジ類をいくつも失っていたのだ。

XC-1498CIIという三菱のCRTがきてからも，黒コゲ君は区役所に引き取られるまで，部屋の隅におかれていた。

学士会館で「きまこん・フォーサイト10周年記念会」が行われたとき，コンピュータマンのササキさんに黒コゲCRTの話をしたら，「4年は早すぎるから抗議したほうがいいですよ」とすすめられた。するとそばからカワムラ先生が，「私は本体が火を吹いたことがあります。あれは怖かったですね」なんておっとりした顔で言うのでおどろいた。

昔，そうじ機や蛍光灯のスイッチを入れたショックで，夫の入力中のデータをすべてとばしてしまったことがよくあった。とても喜劇的な事件なのだが，何かの衝撃でストップしてしまうデリケートさは，コンピュータらしくて，なっとくできる思いがあった。それにくらべると，疲れて火を吹くというアクションは，まったく漫画的ではあるものの，パソコン崇拝で忘れられている電気の怒りのようでドキリとする。

それにしても，「文豪」の墜落，CRTの放電，どうもこんどはPC-9801があやしい。わが家のOA機器軍団は反乱をはじめたのだろうか。

急がばパソコン？

2つのマイコンクラブ合同の10周年記念会は，幹事役の夫の補佐として私も出席させてもらった。オープニングのスピーチでは，それぞれのクラブの創始者が，変化に満ちた10年の思い出を淡々と語ったが，それがかえって印象的だった。

当夜はエヌ・アイ・エフ（NIFTY-Serve）のナカムラ常務をお招きし，「パソコン通信の夢」と題する記念講演をしていただいた。これからパソコン通信はもっと真価を発揮していくであろうし，そのひとつとして，障害やハンデを持った人たちにとって，その有益さが発揮できるものであり

たい。また，ビジネスで長く滞在していたスペインでの実感から，スペイン人とは対照的に，余暇の生かし方のまったくヘタな日本人にとって，パソ通は創造的な余暇を過ごすひとつの希望になるのではないかといったお話だった。

秋は中学校も行事の季節。学芸発表会，区の陸上大会，創立10周年記念式典，日曜参観。もちろん，その間に中間試験，期末試験があり，宿命の内申決定となる。

トオルの予定と夫のスケジュール，それに自分の計画と，3者が関連しあって，秋から冬にかけての日程の調整はとくにむずかしい。気がついたらお歳暮のシーズンになっていたが，20分もあれば行ける日本橋のデパートに，なかなか出かけられない。

その日本橋の老舗（しにせ）デパートのひとつであるT店から，パソコン通信で贈答品の依頼ができる「電子カタログ」が，今年も届いた。1年前の暮れに申し込みをしたとき，無料のソフトがはじめて送られてきた。さっそく利用しようとためしてみたのが，あまりの使いにくさにすぐに見切りをつけて，現場のデパートまで出かけて用事をすませてしまった。それから夏のお中元，今回のお歳暮と，ソフトが送られてきたのは3回目になる。

いままでに，商業通信ネットを通じて文具などのオンラインショッピングをしたことはあるけれど，デパート独自のネットワークを直接利用したことはなかった。それというのも，T店のソフトに失望していたためだが，ほんとにいそがしくて時間がつくれないこんなときこそ，一度はためしてみるチャンスかもしれない。

そこであらためて，T店の「電子カタログ」を手にしてみた。1年前に，グラフィック画面などのスピードの遅さに耐えられなかった記憶があるが，あとはすっかり内容を忘れている。

エンドレスのソフト

3枚のフロッピーディスクは，1枚がシステムディスクで，あとの2枚がデータディスク。ただし，システムディスクには，MS-DOSのシステムと日本語入力FEPが入っていないので，これから始めなくてはならない。

やれやれと思っていたら，マニュアルのいちばん最初に，「マニュアル内容の追加修正がある場合には，システムディスクのなかの README.DOC というファイルに記載してありますので，必ず最初にご覧ください」とある。

指示どおりにすると，修正があるわあるわ……20カ所あまり。しかもファンクションキーの変更など大きな訂正が半分以上で，スタートからびっくりさせる。これを確認したり，プリンタで記録をとったりで何分もかかった。

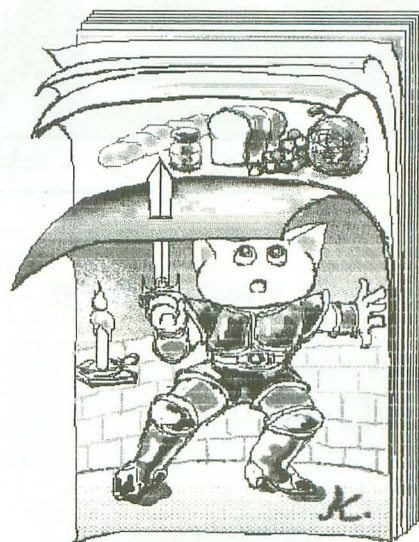
実行ディスクへのインストールのとちゅう，こんどは「必要なファイルのうちコピーできなかったファイルがあります」とメッセージが出るので，やりなおしてみるが同じことがくりかえされる。しかたなく無視してFEPを組み込む。少し強引にやらないと先へすすめない。

これでやっと実行の運びとなる。モデムをONにした状態でディスクを起動。「T店　PCNETWORK」というタイトルと目次がカラーグラフィックであらわれた。これからマニュアルにしたがって，会員登録や商品の注文が始まるはずなのだが，そうかんたんにはいかない。

会員登録の画面では，氏名，性別，年齢などを入力するとちゅうで，どうしても記入できない欄がある。素通りするうち何回目かに，なぜか記入ができる。

商品の一覧では，選んだ画面が不必要とわかっても元の画面にもどることができない。ESCキーやファンクションキーをかたはしから叩いてみてもダメ。何かを選んでつぎに進むことしかできず，「注文」の画面まで連れて行かれる。そこではじめて「発注」か「目次」かを選べるので，元のタイトル画面に帰ることができる。これはたいへんな時間のロスだ。

商品についても，項目だけではわからないから，ちょっと内容を知ろうとしてむやみにキーを選択すると後悔する。のし紙をデザインした装飾画面や，商品のイラストがグラフィックであらわれて，これがけっこうな時間がかかる。しかも前述のように画面のストップや取り消しはいっさいできないのだからまったく困ってしまう。

プログラムを進めていくとちゅう，何回となくハングし，ひんぱんにリセットをくりかえした。四苦八苦して商品を選び出すまでの，ハラハラの長い不安な道のりは，まるでRPGのようだ。

注文の入力もひと波乱だった。単価，数量，金額と，順に記入を終えてリターンをするたび，ひとつ前に入力したものが変化してしまう。いったい正しく記入できているのかどうかわからない。

準備の段階からここまでで，すでに1時間半が過ぎた。地下鉄での日本橋往復，商品をじっさいに見ながらの注文が，頭の中で常に並行している。

まがりなりにも注文の準備が整い，T店へのアクセスとなる。f10キーでオートダイヤルされ，データ送信が行われる。通話はフリーダイヤルなので，当方は無料。

このソフトはT店の依頼で某社が制作したものらしいが，どう考えても習作の段階としか思えない。何人の人がこのソフトを希望したか知らないが，環境設定だけおっくうなのに，走りはじめると事故つづき，カタログの内容はおアソビ同然。天下の日本橋T店はどう考えているのかな。

データ送信が終わるとメニュー画面にもどり，「注文の確認」のキーを押すとふたたびセンターにつながる。きょうのRPGの戦績はどんなものだったのかと画面をみつめると，「本日はメンテナンスのため，サポートセンターにお問い合わせください」と電話番号が示されている。電話をすると時間外のためか応答なし。

おまけがひとつ。最後に通信終了のキーをセレクトすると，いったん回線は切れるが，ふたたびセンターにつながり，モデムをOFFにしないかぎり通信状態は終わらない。

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

高速プリンタ搭載
WD-A351
シャープ

WD-351

シャープは書院ラップトップモデルの最高級機として「WD-A351」を発売した。

「WD-A351」は業界最高水準の82字/秒の高速高精細プリンタを搭載している。また，新たに明朝体，毛筆体，ゴシック体も内蔵の「トリプルスーパーアウトラインフォント」を内蔵しており，明朝に加え，毛筆体，ゴシック体も，名刺に使える小さな文字から垂れ幕やポスターに使える大きな文字まで，104種類の文字サイズで印刷することができる。

そのほかの特徴としては，

・多彩な印刷が可能な「おもしろ印刷」
・手軽に枠組み文書や名刺が作成できるパーソナルDTP
・書院カルクと連動したカード型データベース書院パーソナルカード
・変換効率をさらに高めた約17.6万例のAI-V4辞書
・同社のハイパー電子手帳ともデータの共用が可能な電子手帳通信
・MS-DOSコンバータ，通信ソフトを装備し，パソコンとデータの共用が可能

などの機能を備えている。

価格は238,000円（税別）。

〈問い合わせ先〉
シャープ㈱　☎03(3260)1161,06(621)1221

フルカラーファクシミリ
JX-5000
シャープ

JX-5000

シャープは最大A4サイズまでのフルカラー原稿を高速かつより忠実に送受信することができるフルカラーファクシミリ「JX-5000」を発売した。

「JX-5000」は世界で初めて送受信一体型デスクトップサイズを実現。カラースキャナ分野で培った同社独自のデジタル画像処理技術と新開発の高画質昇華型フルカラープリンタ，ケイディディテクノロジーとの共同開発による独自の画像圧縮伸長技術，さらに専用の高速アダプタなどの要素技術が駆使されている。

送信読み取り部には縮小光学系CCDイメージセンサが採用されており，凹凸のある原稿も読み取りボケなく読み取ることができる。また，一走査でフルカラーを読み取るので色ずれの少ない鮮明なカラー画面データを得られる。

さらに，原稿の拡大/縮小電送を可能にするズーム機能や通信中の画像の乱れを自動修正する自動誤り再送（ECM）機能など各種機能も装備している。

なお，本製品は共同開発先であるケイディディテクノロジーからも同時発売の予定となっている。

価格は3,700,000円（アダプタ「JX-500FX」を含む，税別）。

〈問い合わせ先〉
シャープ㈱　☎03(3260)1161,06(621)1221

音声画像データベース
THE 近江商人
ビットアート

ビットアートから，X68000を利用した音声画像処理データベースシステム「THE近江商人」が発売された。

「いつでも誰にでも使えるコンピュータ」ということで，画像情報処理，そして音声情報も同時に簡単に処理することができるようになっている。

用途としては，

・観光，窓口案内などのディスプレイ
・映像のデータベース
・ポジ-ネガフィルムの管理
・導入研修などの各種マニュアル
・不動産，中古車販売などのデータベース
・各種店頭販売促進ツール

などが考えられ，将来にはISDN化することも可能。

価格はシステムにより，約300万から400万円（X68000とのセット売り）になる。

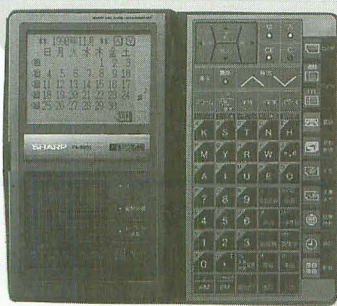
〈問い合わせ先〉
㈱ビットアート　☎0775(52)7190

ハイパー電子手帳用ICカード
PA-9C30/5C01/5C02
シャープ

ハイパー電子システム手帳「DB-Z」用カードとして，3種類のカードが発売される。ラインアップは以下のとおり。

○英和・和英辞典カード「PA-9C30」

辞書の中で最もよく使われる英和/和英辞典を1枚にまとめ，本格的な辞書にも匹敵する約23万語を収録，さらに学習に便利な発音記号も収録している。また，大画面を生かした見出し語リスト一覧表示により，調べたい語をいままでのようにいちいち1語1語順送りしなくても，見たい語にタッチするだけで内容を見ることが可能なので，検索時間を大幅に短縮できる。

綴りのわからない単語のスペルを「？」

PA-5C01

PA-5C02

PA-9C30

で置換し，該当する単語を探すワイルドカードサーチ機能，スペル数がわからなくても最初と最後のスペルを「～」でつなぐだけで該当する単語が探せるブランクワードサーチ機能，そしてビジネス用語，日本文化特色語（和英のみ），慣用表現集（和英のみ），人名，地名の各分野のジャンル別索引なども装備しているので，検索しやすくなっている。

そのほかにも和英辞書で検索した英単語をもとにして，英和辞書と同様に詳しい意味や熟語などをそのまま調べられる，逆翻訳機能などを搭載している。

価格は18,000円（税別）で，2月10日発売予定。

〈問い合わせ先〉

シャープ(株)　☎03(3260)1161,06(621)1221

○「ハットリス」カード「PA-5C01」

ファミコンや業務用ゲーム機で好評の「ハットリス」カード。この「ハットリス」は「テトリス」の作者アレクシィ・パジトノフ氏制作第2弾のゲームである。操作は簡単で，逆さ煙突から落ちてくる6種類のペアの帽子を同じ種類で連続して積み重ねていき，5つ重ねると得点になるというもの。ステージが進むにつれ帽子の種類やスピードが速くなり，鋭い判断力と反射神経が要求される。

価格は6,000円（税別）で現在発売中。

〈問い合わせ先〉

(株)マイクロキャビン　☎0593(51)6482

○囲碁名鑑カード第1巻「PA-5C02」

1989年度に行われた囲碁の7大タイトル戦および話題局など全40局を収録しており，操作しなくても観戦できる「自動」モードと，1手1手確認しながら観戦できる「手動」モードの2つのモードで実力を持った棋士同士の熱戦を手軽に電子手帳上に再現する。

また，収録された対局の全戦で「次の1手」を推察して打ち，その正否で棋力を判定する「判定」モードで実力を試すこともできる。中級では5つのヒントの中から上級はノーヒントで棋士の打った次の手を選ぶ。1手ごとに正否がわかるほか，対局終了後に実力が画面に表示される。7,000円。

〈問い合わせ先〉

(株)ヘクト　☎03(5275)5481

CCITT V.42bis搭載

MD96FS5V

オムロン

オムロンは「MD96FS5」をバージョンアップした機種，「MD96FS5V」を発売した。

「MD96FS5V」は，あらたに国際標準データ圧縮機能としてCCITT（国際電信電話諮問委員会）が昨年勧告したCCITT V.42bisを標準搭載している。

主な特長は以下のとおり。

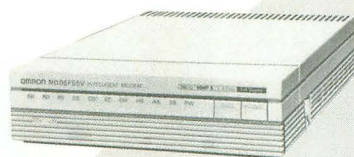
・国際標準通信規格であるCCITT V.32をサポートしているので，本格的な9600bps全二重通信を実現し，V.32を搭載したモデム同士で通信が可能。

・エラー訂正機能としてMNPクラス4とCCITT V.42を標準搭載している。

・データ圧縮機能にMNPクラス5とCCITT V.42bisを標準搭載しているので，ソフトなどの工夫なしで実効通信速度が19200bpsまで向上。

価格は198,000円（税別）。

〈問い合わせ先〉

オムロン(株)　☎03(5488)3219

MD96FS5V

コモドール

AMIGA内蔵CDROM

CDTV

コモドールの新しい世界戦略マシンである「CDTV」（Commodore Dynamic Total Vision）はAMIGAとCDROMを結合させた，まったく新しいカテゴリーのコンピュータである。「CDTV」のいままでのCDROMと異なる点は，一見普通のCDプレイヤーと思われる本体の中に，AMIGA500（＋512Kバイト RAM）相当のコンピュータが内蔵されていることであり，そのうえ1,000ドル程度と安価になっている。

標準ではキーボードやマウスなどは付属しておらず，すべての操作は付属のIR（赤外線）リモートコントローラによって行うことが可能。このリモートコントローラはいままでのマウス，ジョイスティック，キーボード，CDROM操作のすべての機能を搭載しながらも使いやすくなっている。

主なスペックは以下のとおり。

・CDROM部

データ読み出し　：153Kバイト/秒 MODE1
　　　　　　　　：171Kバイト/秒 MODE2
　　　　　　　　：2Mバイト/秒 burst

平均アクセスタイム：0.5秒

最大アクセスタイム：0.8秒

Commands　：CDROM、CD-Audio、CD＋G

MTBF　：10,000 P.O.H.

CDTV

Standard Supported　：ISO-9660

容量　：540Mバイト

・入出力端子

アナログ，デジタルRGB（DB-23）出力

コンポジットビデオ（NTSC）出力

S映像出力端子

RF出力

セントロニクスパラレルインタフェイス

RS-232Cシリアルインタフェイス

外部フロッピーディスクドライブ

ステレオオーディオ出力

MIDI IN/OUT

RAM/ROMカード（256Kバイト）ポート

現在のところは欧米などでしか発売されていないが，今年春ごろから日本でも発売される予定。

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。寒さもいちだんと厳しくなってきた今日この頃。でも春はもうすぐそこ，何事にもめげずに頑張ってくださいね。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
ポケコンジャーナル　工学社
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶年末年始パソコン購入ガイド'91
Macintosh やAMIGAといった洋モノからPC-9801，X68000，FM TOWNSといったお馴染みの機種まで，各マシンの特徴やコストパフォーマンスを解説。購入予定のある人は必読。——編集部，LOGIN，24号，251-263pp.

▶NETWORK CONNECTION
ネットワーカー・ホリックが新装開店。今号では，EYE-NETの情報誌とAV専門誌が運営する「aVle」を紹介。X68000のPDSはDōGAが開発したウィンドウシステム「KO-WINDOW」。——編集部，LOGIN，24号，296-297pp.

▶Goods Collection
シャープから発売された電子手帳PA-8800とPA-6300の紹介。——編集部，ポケコンジャーナル，1月号，1p.

▶日本パソコン百景
日本を飛び出して台湾の台北中華商場へ。いわば台湾の秋葉原。その混雑と汚さにビックリの取材陣。——フデヨシ＆カシワラ，ASCII，1月号，286-288pp.

▶サイバーソン（CYBERTHON）報告
サンフランシスコで行われたバーチャル・リアリティに関するイベント，サイバーソンの模様を詳細にレポートする。——野々村文宏，ASCII，1月号，369-376pp.

▶MIT Media Lab.
MIT Media Lab.創設5周年記念シンポジウムの2日間の模様を，コボレ話も含めて紹介する。——砂原秀樹，ASCII，1月号，433-437pp.

▶ASCII EXPRESS
ラスベガスで開催された世界最大のコンピュータトレードショー，COMDEXの模様をレポートする。——編集部，ASCII，1月号，282-283pp.

▶グローバル・ビレッジ・ダイジェスト
世界最大のコンピュータショウ，COMDEXについて，その性格から内容，日本に対する影響まで幅広くレポートする。——高田正純，マイコン，1月号，140-149pp.

▶各種モデムの性格診断
X68000を使って行われたモデム7機種のスピードテストを豊富なグラフによってまとめる。——マイコンネットワーク研究会，マイコン，1月号，150-158pp.

▶パソコンソフト・マーケットガイド
ホビーソフトの年間ベスト10と，お勧めのゲーム20本を紹介。——編集部，マイコン，1月号，別冊46-70pp.

▶秋季コムデックス
COMDEXの模様を主に会社別に紹介していくレポート。——Dana Blankenhorn，I/O，1月号，238-241pp.

▶アナログ回路による重低音システムの設計
カーオーディオに重低音を追加する装置を設計する。3Dフィルタとパワーアンプを使ったもので，ヘビメタ愛好者には必需品。——山崎誠，I/O，1月号，196-204pp.

▶I/O総目次
I/O版INDEX'90。1年間のI/Oの記事を手早く探せる総目録。——編集部，I/O，1月号，214-216pp.

MZシリーズ

MZ-1500（MZ-5Z001BASIC）

▶漂流
海を漂流しているロジンソン。飢え，渇き，絶望に耐えながら陸地を目指すゲーム。——ヘルニャン・バウマン，マイコンBASIC Magazine，1月号，122-124pp.

MZ-2500（M25-BASIC）

▶恒例！　町内玉入れ大会
みこちゃんを操って，玉をすべてゴールのカゴに入ればOK。パズルゲーム。——謎のパズル大好きおじさん，マイコンBASIC Magazine，1月号，125-126pp.

X1/turbo/Z

X1シリーズ

▶ドロアーガの塔　エクサラント
各階のボスやスライムと戦い，宝を取り，罠や迷路をくぐりぬけ，泉で休み，姫を無事に救け出す。——堀田英克，マイコンBASIC Magazine，1月号，151-152pp.

▶ヘズメタルジャケット
2～6人用対戦戦車ゲーム。対戦バトルシティ。要ジョイスティック2本。——ヘミタク，マイコンBASIC Magazine，1月号，153-154pp.

X1turboシリーズ

▶A STARRY NIGHT　～星の多い夜～
洗脳された星をブラック・ホールに沈めろ。——中西弘幸，マイコンBASIC Magazine，1月号，155-157pp.

X68000

▶NEW SOFT
生中継68，Misty Vol.6，Magical Shotを紹介。——編集部，LOGIN，23号，18-27pp.

▶最新ゲーム徹底解剖!!
新着ゲーム「アトミック・ロボキッド」「ソル・フィース」「ニューラル・ギア」とRPG「ラグーン」を攻略。——編集部，LOGIN，23号，170-191pp.

▶よくわかる続ダンジョン・マスター カオスの逆襲
開発中のシステムを機種ごとに紹介。——編集部，LOGIN，23号，200-201pp.

▶Software Review
超難解シューティング「ナイアス」をサカナに，アクション・シューティングの近未来を語り合う。——松本

新刊書案内

NTTが核攻撃される。どこが攻撃するのか。ソ連である。どうして攻撃するのか。アメリカの極東核戦略において，重要な国防通信システムの一端を担っているからである。なぜ攻撃されるのか。通信網がやられれば現代の戦争は戦えないからである。在日米軍基地にアメリカの通信システムがあるのは当然だが，米軍とはまったく関係のないNTT国分寺電話局や，KDD大手町電話局も米軍の軍事通信に使われているのだ。

と，こういうドキュメントである。ウサンくさそうなタイトルだが，米軍が日本からも核攻撃指令ができるようなシステムをとっていても不思議はないし，資料もきちんと揃えてある。ポイントは日本が国民の知らされないまま，重要な核攻撃の拠点になっていることや，日本の光ファイバー技術がひと役かっていること。民間企業であるはずのNTTやKDDが米軍の協力をしてケーブルを引いたり回線を用意したりし，日電が米軍のNORADと近くの指令部を光ファイバーで結ぶ際，光通信装置を納入した。とかく見えないところで物騒なのは次戦緩和のいまもかなりないのだ。　（K）

NTTが核攻撃される日　浅井隆著　フットーワーク出版社刊　☎03(5395)5711　四六版　295ページ　1,500円

書籍の価格は消費税込みです

隆一VS. X68000新聞社，LOGIN，23号，208-209pp.
▶X68000新聞
イメージファイト，ソル・フィース，ブルトン・レイ，マーブルマッドネス，スペース・ローグを紹介。——編集部，LOGIN，23号，268-271pp.
▶NEW SOFT
ルーシー・ショット，ブルトン・レイを紹介。——編集部，LOGIN，24号，28-29pp.
▶The News File
SUPER-HDのハードディスクのないタイプ，「X68000 SUPER」を紹介。——編集部，LOGIN，24号，39p.
▶最新ゲーム徹底解剖!!
シューティングゲーム「ソル・フィース」の6面を徹底解剖。——編集部，LOGIN，24号，200-201pp.
▶X68000新聞
遥かなるオーガスタ，ワールドスタジアム，栄冠は君に，リングマスターII，プリンス・オブ・ペルシャ，を紹介。また，発売が再び増え始めたシューティングゲームを「第二次シューティングブームだ!!」と銘打って特集。——編集部，LOGIN，24号，274-279pp.
▶パソコンゲーム大全
「銀河英雄伝説II」と「続ダンジョン・マスター カオスの逆襲」の攻略法。——編集部，コンプティーク，1月号，136-139，160-161pp.
▶GAMING WORLD
イメージファイト，ジェミニウイング，ソル・フィース，ニューラル・ギア，マーキュリー，ノスタルジア，シュヴァルツシルト，リングマスターII，エメラルド・ドラゴンの紹介。——編集部，テクノポリス，1月号，38-57pp.
▶攻略ファイト！
「続ダンジョン・マスター カオスの逆襲」と「銀河英雄伝説II」を攻略。——編集部，テクノポリス，1月号，80-83，88-89pp.
▶Hot Press
続ダンジョン・マスター カオスの逆襲，ブルトン・レイ，ニューラル・ギアの紹介と攻略法。リングマスターII，生中継68，ワールドスタジアム，ソル・フィース，イメージファイトの紹介。——編集部，POPCOM，1月号，16-32pp.
▶ゲームの達人
銀河英雄伝説IIを攻略。——編集部，POPCOM，1月号，90-93pp.
▶ミュージック・パビリオン
X68000 X-BASIC用ミュージックデータ「TRUTH（T-SQUARE）」。——編集部，POPCOM，1月号，183-187pp.
▶NAGDRV情報局
マイコンBASIC Magazine1990年12月号で発表されたミュージックドライバソフト「NAGDRV」についての，読者からの質問に答えている。——永田英哉，マイコンBASIC Magazine，1月号，88-89pp.
▶誌上公開質問状
X-BASICで64KBを超えるADPCMの録音のやり方は？——多田太郎，マイコンBASIC Magazine，1月号，92p.
▶NEW PRODUCTS
グラフィックソフト「CANVAS PRO-68K」の紹介。——編集部，マイコンBASIC Magazine，1月号，95p.
▶ブラディオン
宇宙船を操って相手をやっつける。1～2人用対戦シューティング。——片岸健一，マイコンBASIC Magazine，1月号，158-159pp.
▶MENTAL POWER
向かってくる壁を左右に避けて進み，決められた数の壁をよけると次のステージへ。ドライブゲーム。——高橋秀之，マイコンBASIC Magazine，1月号，160-162pp.
▶UP AND DOWN
海底で潜水艦を動かし，お金を取る。——永井崇博，マイコンBASIC Magazine，1月号，163p.
▶WGP　～Name Entry～
タイトーのゲームミュージック・プログラム。——西島淳一，マイコンBASIC Magazine，1月号，183p.
▶ATMIC ROBO－KID　～メインテーマ～
ゲームミュージックプログラム。要NAGDRV。——あんど，マイコンBASIC Magazine，1月号，187-189pp.
▶FINAL LAP2　～EndingA～
ナムコのゲームミュージックプログラム。要NAGDRV。——Hideya Nagata，マイコンBASIC Magazine，1月号，190-194pp.
▶チャレンジ!!　アドベンチャー・ゲーム
「闇の血族・完結編」を，画面写真で紹介。——佐久間亮介，マイコンBASIC Magazine，1月号，221-223pp.
▶ソル・フィース
シューティング「ソル・フィース」を紹介。——山下章，マイコンBASIC Magazine，1月号，226-227pp.
▶ゲーム虎の穴
「X68000はゲーセン移植がすごい」と称してサイバリオン，グラナダなどのゲームとMIDIインタフェイスを取り上げる。——編集部，ASCII，1月号，321-336pp.
▶AV　STRASSE
FIXER Ver.4.0，ドローグラフィックライブラリなどの市販ソフトと，KO-WINDOW，ファイル圧縮ツールLZXなどのPDSを取り上げ，評価する。——仲田津宏・中山進，ASCII，1月号，409-412pp.
▶FREE SOFTWARE INDEX
PDSの中から目立ったものをリストアップ。X68000用BGC，DRV，JSHWILDなども紹介されている。——編集部，ASCII，1月号，451-455pp.
▶光磁気ディスクCZ-6MO1をレポート
X68000ユーザー待望の大容量メディア，MOディスクの概要と使用感のレポート。値段，互換性，活用法，付属ソフトなどについて詳細に取り扱っている。——高橋雄一，マイコン，1月号，281-285pp.
▶X68000マシン語入門
今月からは対話型ソフトの製作を行う。第1回はキーボードからの1項目入力ルーチンを作るところまで。——高橋雄一，マイコン，1月号，292-299pp.
▶ハードディスクIPL
ハードディスクがついているけどフロッピーからの立ち上げ機会が多い人に便利なフロッピー優先のIPLプログラム。——市原昌文，I/O，1月号，172-179pp.
▶DELBAK2
ソースなどのバックアップファイルをまとめて消去するプログラム。——L＆M，I/O，1月号，205-206pp.
▶GAME BOX
アトミック・ロボキッド，ナイアスについて批評をする。——吉沢正敏・牛島健雄，I/O，1月号，132-133pp.
▶SOFT BOX
シャープから発売されたCANVAS PRO-68Kで，ドローツールの仕組みから絵の書き方，使い道などについて紹介する。——伊藤ゆう，I/O，1月号，210-211pp.

ポケコン

PC-E500
▶ATTACK BALL
ボールを相手に当ててやっつける。——町野稔，マイコンBASIC Magazine，1月号，166-167pp.
PC-E500PJ
▶誌上公開質問状
「μ」や「Ω」といった記号を表示するには？——Akiko，マイコンBASIC Magazine，1月号，91-92pp.
PC-E500/PC-G801
▶ポケコンQ＆A
PC-E500でのKEY0の使い方や乱数の発生法，PC-G801でのダンジョンマップデータの展開方法などの質問について答える。——編集部，ポケコンジャーナル，1月号，82p.
PC-E500/E550/1480U/1490U
▶はみだしゲーム講座
仮想VRAMを使った背景の処理の方法について説明する。——編集部，ポケコンジャーナル，1月号，87p.
▶FRUITS FIELDS
幻の秘宝「ふるうつ」を集めて回るパズルゲーム——r・emoo，ポケコンジャーナル，1月号，13-20pp.
▶2L・PYONKO！
穴に落ちないように左から右へ跳ぶアクションゲーム。——せとけん，ポケコンジャーナル，1月号，48p.

幻想としての文明

ニッポンの終焉や縄文式思考云々に次ぐ，栗本慎一郎教授の一般人向け啓蒙書である。氏がいままで書いてきたさまざまな論理や事象を中心に時事ネタを交えて述べた書であり，学術書ではない。「文明の旋律理論は，物的構造がすべてであるという誤った理論を完全に廃止する。たとえば，餓えた者は立ち上がり，豊かな者は不満を持たないというような『理論』はもう捨て去るべきことを主張する」本だ。（K）

栗本慎一郎著　講談社刊　☎03(3945)1111　四六版　200ページ　1,300円

電脳と頭脳

原題を訳すと，「アイデアと情報」。人間の知能とコンピュータのもたらす記号処理を分離し，人間がコンピュータに過剰に期待したり，恐れたりせずにつきあっていけるよう書かれた本である。頭脳はアイデアを生み出すことができるが，電脳にはできない。本書はそれを骨子に，コンピュータテクノロジーの概念とそれと正しくつきあうための知識や知恵を平易に解説している。（K）

アーノウ・ペンジアス著　岡和夫訳　TBSブリタニカ刊　☎03(3238)5711　四六版　266ページ　2,000円

書籍の価格は消費税込みです

QUESTION and ANSWER

X-BASICでゲームプログラムを組んでいます。スプライトとBGを使っているのですが，BGを画面にしきつめてしまうとテキストが見えなくなってしまいます。スコア表示もしたいし，実行中にブレイクキーを押したりエラーが出て止まってしまってもどこで止まったかもエラーの内容も見えないので非常にうざったいのです。

どうにかしてBGを使いながらテキストを見られるようにしたいのですが，そういうことはできないものなのでしょうか。

神奈川県　富良圭介

通常X-BASICでは画面についてはスプライト（BG），テキスト，グラフィックの順に優先順位が決まっていて，BGがしきつめられているとBG（スプライト画面）の裏にテキストの表示が隠れてしまって表示することができません。

ですがX68000は実はマシン語レベルではその優先順位が変えられるようになっています。プログラム中はctrl＋Dを押して画面を初期化するというのがふつうですが，この際ですから画面の優先順位を変える拡張関数を作ってしまいましょう。

X68000の画像は専用のビデオコントローラによって複数の画面を合成して表示データをドット単位に送り出しています。アドレスでいうとE82400$_H$〜E82600$_H$がそうでここにいろいろな数値を書き込むことでテキスト/グラフィック表示のON/OFFを行ったりその優先順位を決定することができるようになっているのです。そしてE82500$_H$番地がその優先順位を決定するI/Oでここに数値を書き込むことでテキストをスプライト画面の上に表示させるようにします。

E82500$_H$には初期状態では2進数で表現すると次のようなデータが書き込まれています。

00　00　01　10

このうち上位2ビットは無意味，次の2ビットがスプライト画面の優先順位（0番目。つまり一番上），次がテキスト（1番目）でグラフィック（2番目）という内容になっているのです。ここを，

00　10　00　01

と書き直すことで画面の優先順位をテキスト，グラフィック，スプライト（BG含む）という順番にすることができるわけです。

ということでリスト1がそのサンプルです。asx, lk.xでアセンブル，リンクして作成された .Xファイルを .fncという拡張子にリネームしてBASICが入っているディレクトリに転送します。

次にBASIC.CNFというコンフィギュレーションファイルに，

FUNC = ファイル名

という1行を追加してください。これでtxtup( )関数がX-BASICに追加されます。この関数は引数に0を入力するとテキストがBGの上に，それ以外の数値でBGがテキストの上に表示されるようになります。

ビデオコントローラに画面の優先順位を設定する数値を入力する際に注意しなくてはならないのは異なる画面（テキスト，グラフィック，スプライト画面）にそれぞれ違う数値を入れなくてはいけない（つまり複数の画面に同じ順位をつけてはいけない）ということです。そのためユーザーが変な数値を入れないようにこのプログラムではテキストを上にあげるだけになっていますが，67/72行目の数値を変えるだけでグラフィックを一番上にしたりスプライトを一番下にしたりといろいろと画面の設定を変えることができます。プログラム中にいろいろとコメントも残しておきましたのでぜひ各自好きなように変更して見てください。

また，1月号でディスクに収録されていたMUSICDRV.X用のMUSIC1.FNCには，画面の優先順位を変更する関数も入っています。詳しくはドキュメントをお読みください。

リスト

```
 1: *  テキストVRAMの優先表示
 2: *    By DE 1990/11/23
 3: *    .include iocscall.mac
 4:
 5: *     Information Table
 6:        .text
 7:     .dc.l   F_init
 8:     .dc.l   F_run
 9:     .dc.l   F_end
10:     .dc.l   F_exit
11:     .dc.l   F_break
12:     .dc.l   F_ctrlD
13:     .dc.l   F_dmy1
14:     .dc.l   F_dmy2
15:     .dc.l   F_token
16:     .dc.l   F_parTbl
17:     .dc.l   F_exec
18:     .dc.l   0,0,0,0,0
19:
20: F_init:
21: F_run:
22: F_end:
23: F_exit:
24: F_break
25: F_ctrlD:
26: F_dmy1:
27: F_dmy2:
28:     rts
29:
30: *     TOKEN table
31: F_token:
32:     dc.b   'txtup',0
33:     dc.b    0
34:
35:        .even
36:
37: *     param table
38: F_parTbl:
39:     dc.l     txtup_par
40:
41: *     parameter ID table
42: txtup_par:
43:     dc.w   $02         *引き数int(省略不可)
44:     dc.w   $ffff           *戻り値なし
45:
46: *    EXEC TAble
47: F_exec:
48:     dc.l   txtup_func
49:
50:     .even
51:
52: *    エントリ
53: txtup_func:
54:     tst.l    12(sp)
55:     bne   gotxtdn
56:     bsr   txtupset
57:     bra   retfunc
58: gotxtdn:
59:     bsr   txtdnset
60: retfunc:
61:     clr.l   d0
62:     rts
63:
64:
65: *    優先順位セット（テキスト優先）
66: txtupset:
67:     move.w  #%00_01_00_10_11100100,d1
68:     bra   writeit
69:
70: *   （スプライト優先）
71: txtdnset:
72:     move.w  #%00_00_01_10_11100100,d1
73:
74: *    値を書き込む
75:
76: writeit:
77:     move.l    #$e82500,a1
78:     IOCS     _B_WPOKE
79:     rts
80:
81:     .end
```

12月号Q＆A，179ページのバッチファイルを入力して起動したら，登録してあるファンクションキーがクリアされてしまい，再登録しなければなりませんでした。なぜでしょうか。またクリアしないバッチファイルにするには，どうすればよいのでしょうか。あと，Bドライブにある OPMファイルを連続演奏するバッチファイルを作ろうといろいろやってみましたが，うまくいきませんでした。曲が終了したことを返すコマンド，あるいはエラーコードを返すツールがないものでしょうか，ご教示ください。

また，さらにBドライブのファイルをランダム演奏するにはどうしたらよいのでしょうか。使用機種はX68000ACEです。

宮城県　佐藤　公夫

たくさん質問がありますが順番に答えていきましょう。まず最初の質問ですが，これは佐藤さんが本文の説明をよく読んでいないんだと思います。ほかにも何人かの方からそういった質問を受けましたが，編集室で動作を再確認したところ，ファンクションキーがクリアされるといった症状は見られませんでした。12月号をよく読んで再挑戦してみてください。おそらく，“_exit(i)”と書き換えるところを“b_exit(i)”としているんだと思います。

次の質問は，バッチファイルの中で曲の終了を教えてくれるコマンドがあるのか？ということですが，そのようなコマンドはありません。しかし，あきらめるのはまだ早いですよ。曲の終了を返すコマンドはありませんが，指定したチャンネルが演奏中かどうか調べる命令がX-BASICにあります。これが“m_stat”です。マニュアルにあるように引数を省略すると，チャンネル1～8が演奏中かどうか調べてくれます。

このときの戻り値は，ビット0～ビット7がチャンネル1～8に対応していて，0で停止中，1で演奏中を示します。つまり，戻り値が0なら曲が演奏されていないということです。簡単な実験をしてみましょう。

```
10 m_init( )
20 m_trk(1,"o4cdefgab<c")
30 m_play( )
40 repeat
50 until m_stat( )=0
60 print "演奏が終了しました"
70 end
```

見てのとおり「ドレミファソラシド」を鳴らすだけのつまらないプログラムですが，実行させると演奏が終了してからメッセージを表示するようになっています。仕掛けは40行からのrepeat～untilです。m_stat( )の戻り値が0（つまり全チャンネルが停止）になるまで，ループを繰り返すようにしてあるのです。

これで演奏の終了判定のやり方がわかりましたが，このままではバッチファイルから使えません。そこでまたまたCコンパイラを使うことにしましょう。下のプログラムをX-BASICから入力してください。ファイル名はwait.basとしましょう。

```
10 repeat
20 if inkey$(0)<>"" then break
30 until m_stat( )=0
40 end
```

ループの中でキー入力を見るように変更したので（20行），演奏途中でもなにかキーを押せばループを抜けるようになっています。次にBC.XでC言語のプログラムに変換したら，エディタに読み込んで“b_init( )”という行を削除して，“b_exit(0)”を“_exit(0)”と書き換えてください。こうしたら，

```
cc wait.c /W
```

としてコンパイルします。オプションの“/W”は必ず大文字で指定してください。これでwait.xが生成されます。

```
echo off
copy b:a.opm opm
wait
copy b:b.opm opm
wait
 ⋮
echo on
```

というようなバッチファイルを作ってみてください。a.opmの演奏が終了するか，なにかキーを押すと，b.opmの演奏が始まるはずです。これでOPMファイルの連続演奏ができるようになりました。しかし，上のバッチファイルだとOPMファイルを追加したときに，バッチファイルを書き換える必要があります。それでは面倒なので，

```
echo off
for %a in (b:*. opm) do copy b:%a opm || wait
echo on
```

としておけばバッチファイルを書き換える必要はありません。上の“||”はBASICの“:”のようなもので，1行に複数のコマンドを列挙するときに使うものです（“|”はシフトキーを押しながら“¥”を押して入力します）。

もしも，OPMファイル自体に手を加えることも辞さないならば，もっと簡単な方法もあります。OPMファイルの最後，“(p)”となっている部分の次に“(w)”を加えればいいのです。これは演奏が終了するまでウエイトをかける命令です。これなら，

```
COPY *.OPM OPM
```

だけで全曲演奏ができます。ただし，変更したOPMファイルはバックグランド実行できなくなりますので注意してください。

それとBASICなどでプログラムを作り，コマンド化すればランダム演奏もできないことはないはずです。ぜひ佐藤さん自身で考えてみてください。

最後にOPMファイルをOPMドライバにコピーするときに注意しておきたいことを書いて終わりにします。普通OPMドライバを登録するシステムのCONFIG.SYSは，

```
DEVICE=¥SYS¥OPMDRV.X
```

のようになっていると思いますが，このとき確保される演奏用バッファは約64Kバイトです。ですから最近では11月号に掲載された「GUSH」のように，1トラックに10Kバイトとるようなプログラムだと，X-BASICが「外部関数エラーです。メモリー確保できません」とエラーメッセージを出すでしょう。X-BASICは親切にメモリが確保できなかったことを教えてくれますが，OPMドライバに直接コピーして演奏する場合にはそんなメッセージも表示されません。メモリが確保できなくても演奏はされますが，原曲を忠実に再現してはいないのです。そこで演奏用バッファを多めに取っておけばいいのですが，それには，

```
DEVICE=¥SYS¥OPMDRV.X #80
```

のように#に続けて数字を書きます。上の例だと演奏用バッファに80Kバイト確保されます。そんなことは知らずにCOPY～OPMしていた人もいると思うので，思い当たるフシのある人は，CONFIG.SYSの設定内容を確認してみることをお勧めします。

（影山裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

さて，いよいよ大詰め。何がって？　決まっているじゃないですか。ね，受験生の皆さん。試験が始まるまでは自由に勉強していいんですから，存分にやってください。試験が始まってから，参考書見たりしたらだめですよ。

◆ゆ，雪が降らない。このままじゃ，スキーに行けない。夏にどこにも行かずにバイトしたのに。こうなったら，スキーへ行くのを1回やめて，C compiler PRO-68K ver.2.0を買っちゃおうかな。ところで，いまうちのSX-WINDOWでは2種類の暁子さんが走っている。旧バージョンのSX-WINDOWを持っているそこのあなた，新しい名前にして同じディスクに入れてあげて，MacとX68000の両方を楽しみましょう。しかし，マウスカーソルがあんな形になるとは……（暁子さんのウィンドウの中で骨付き肉や魚になってしまうとはね）。　大内　泰司(21)千葉県

スキーとC compiler PRO-68K ver.2.0を天秤にかけるというのも，なかなか妙な感じですね。

◆12月2日，日本人初の宇宙飛行士としてTBSの秋山さんが宇宙へ行った。しかし，その第一声がすごかった。「これ，本番ですか？」（さすが，TBS関係者だけのことはある）。

神生　直敏(21)栃木県

しかし，宇宙に行けるというのは本当にうらやましいですね。虫歯があるとだめらしいからなあ。

◆4歳の孫が来ると，X68000は孫のオモチャとなり，私は助手にされてしまいます。Z'sSTAFFでのお絵描きは得意中の得意，源平討魔伝，ポピュラスにのめり込んでいるという，末恐ろしい男の子です。孫のために（口実）CANVAS PRO-68K購入の予定です。　中野　譲(64)兵庫県

「孫のため」といいわけすれば，なんでも買えそうな気がしますね。でも，誰にいいわけするのかなあ。やっぱり，奥さん？

◆いつもごくろうさまです。ぼくたち中学生にもわかりやすい記事などをお願いします。

遠藤　塁(14)岩手県

いやあ，なんかよくわからないけど，すがすがしいハガキだなあ。

◆まったく，高校での修学旅行はどこへ行くのかと思えば，ワンパターンの京都，岡山。ま，これはしかたがないとしても問題はいついくかだ。1月30日とは何を考えているんだか。とはいっても，せっかくの修学旅行で1日中大阪日本橋に行く私のグループも何を考えているんだかわからんが（先生があまりにもうるさいので一応大阪城にもよる予定）。

清水　勲(16)神奈川県

関東のほうはやっぱり修学旅行といえば大阪，京都になるんでしょうね。最近はお金持ちの学校になると，海外に行くのも当たり前になっているようですけど。

◆1校すべって腹いせにX68000で遊んだ。メモリをダンプすると“浪”なんて書いてある。ムカーッ，X68000までバカにしてやがる。次の日も1校すべっていた。受験の日に母さんが「ヘビが落ちる夢」を見たそうな。やっぱり，“浪”なのだろうか（実話）。　渡辺　篤志(17)滋賀県

しかし，お母さんも受験の日の朝に「ヘビが落ちる夢を見たわよ」とは口が裂けても言えなかったでしょうね。

◆少し前のことだが，学校でC言語の講習があるというので行ってみた。もちろん，XCではなくPC-9801上で動いているTURBO Cだった。おもむろにプログラムを打ち込みSAVEしようとした。データディスクも確認し，実行した。しかし，SAVEできない。しばらく悩んでいると友達がライトプロテクトになっているといってくれた。そうです，3.5インチディスクではプロテクトノッチが開いていると，プロテクトがかかるのですね。5インチしか使わないから知らなかった。　浅沼　博明(20)北海道

3.5インチディスクのプロテクトノッチはいいですよね。5インチもあんなふうにできないのかなあ。

◆ただいま卒業研究でシミュレーションプログラムを暇プロ的に作っています。そういう理由で「清水和人流プログラミング道場」を心待ちにし，「シミュレーションプログラミング入門」の今後の活躍に期待している今日この頃です。

藤戸　正道(22)東京都

卒業研究ですか。いまのうちは暇プロ的にやっていても，せっぱつまってくるとそうもいかないでしょうね。

◆バスの中でOh!Xを読んでいて，友人に「この本，560円の価値あるの？　なんか内容が少なそう」などと言われた。僕は猛烈に悲しかった。こんなにためになる本はないと思うのに。

佐竹　勝博(16)香川県

うーん，それは猛烈に悲しい。

◆突然ですが，私の通っている学校には無数にあるPC-9801，ポツポツとあるワークステーションに「こそっ」と隠れるように備品票のついたX68000 PROII-HDがあるのです。一時はサイバースティックもあったのですが，教授に送り返させられたそうです。1階下にあるその研究室では「R-TYPE」マシンと化したX68000の姿が21インチのディスプレイと共にあったのですが，最近は修論のために使われているみたいです。12月ですからね……。ところで，私は現在48時間以上起きているのですが眠くならないのです。どうしたことだ……，へこへこ。

折田　正栄(22)福岡県

ついに，人類永遠の課題といわれる眠らない術を身につけたのでしょうか。

◆誕生日に両親に髭剃り（電動のやつ）をもらいました。いままでは手で毎朝ジョリジョリやっていたのですが，これはいいですねえ。ズームイン朝を見ながらブィーンと剃っています。

◀伊藤　浩克　香川県
これはイースの絵ですね。1つひとつの線がはっきり書かれているので，全体的にきれいに，また，きっちりと見えますよね。

◀松本　浩幸　大阪府
これもきれいでかわいい絵です。手の位置がよくわからないような気もしますが……。それにしてもみんなF-1のゲームに飢えているようですね。

手でやるよりずっと早いですね。しかし，人間はこうやって文明の利器の中に埋もれていくのですね。ときどきは手で削ろうかな。あ，プレゼントしてもらったのはブラウンのやつです。「おはようございます。ブラウンです！」のあれです。　藤田　勝也(22)神奈川県

文明の利器はなんといっても便利だし，時間の節約にもなりますからね。埋もれてしまってもいいんじゃないでしょうか。

◆メジロマックイーン，キョウエイタップ，パッシングショット，単勝の3連勝は我ながらすごいです。しかも，1点勝負ですよ。でも，千円ですからね。あまり運を使うとあとが怖いし，有馬記念はじっとしておくことにしましょう。天皇賞のバンプーは本当に残念でした。ホワイトストーンがジャパンカップで勝ったら，C compiler PRO-68K ver.2.0に化けるかも。

藤沢　直樹(20)大阪府

競馬の馬の名前ってなんか変。

◆うちの高校の合唱部が都代表となり，北海道で行われた全国大会に行ってきました(全国"2位")。コンピュータルームのとなりが音楽室なので毎日全国レベルの歌を聞かされています。いまでは歌も覚えてしまい，一緒に歌っています。でも，プログラムを打っているときはうるさくてむかつきます。そして，なにを隠そううちのクラブもDōGAのコンクールで入選('88と'89年)しているんですよ，これが。

市川　徳明(17)東京都

コンピュータで演奏させて，それに合わせて合唱してもらうというのもいいんじゃない？

◆この前の夏休みのことであるが，私はアルバイトで地下探査アシスタントというものがあったのでやってみた。しかし，実際にはなんと自衛隊の演習場で不発弾探しをやらされた。しかも，そこはハチは飛ぶわ，マムシは出るわ，挙げ句の果てにはクマも出たことがあるということを聞かされたのであった。まったく，こんなバイトをした人は数少ないであろう。

山本　潮(21)宮城県

クマまで出るとは。自衛隊の人もときどきはクマに遭遇するのかな。

◆日本サンライズはサンライズに。日本ソフトバンクはソフトバンクに。でも，もし日本食堂が食堂になるとなんか変!?

黒武者　健一(21)神奈川県

変というよりも，無茶苦茶情けない。絶対やらないでしょけど，やったら面白いのになあ。

◆最近見かけないので，てっきり事故って入院でもしているのかと思っていましたが，元気だったんですね，華門さん。私は先日，自転車で山道を下っていて，カーブを曲がり切れずに崖(といっても4mぐらい)から落っこちました。幸い下の地面が柔らかかったので，肩がはずれたほかは打撲だけですみました。編集部の皆さんも運転するときは気をつけてください(それにしても落ちていくときのあの感覚は病みつきになりそうである)。　中村　学(19)石川県

4mだと結構落ちるまでの時間が長いかもしれない。いままでの人生が走馬灯のように浮かばなかったでしょうか。

▲小川　裕美　山口県
先月も載せた人なんですが，また載せてしまいました。いよいよ女の子をひいきしてるようですが，投稿の数が少ないということもあるんですよね。

▲岩瀬　貴代美　福岡県
この人も連続で載っています。うまいですよね。しかし，左のような事情もありますので，皆さんどんどんイラストを投稿してきてください。

◆昔々の話です。私はOh!MZに掲載されているプログラムをX1で打ち込もうと電源を入れました。そのとたんに，なぜか私はサザエさんが見たくてたまらなくなりました。うまい具合にあと30分でサザエさんが始まる時間だったので，私はX1のタイマーをセットしてからBASICを立ち上げました。で，問題の時間に画面は切り替わるはずだったんですが……，なぜか画面は切り替わらずに真っ青に染まったX1の画面が映っているのみでした。ああ，哀れX1は修理に出されたのでした。それにしても恐るべしサザエさん。不老不死はダテじゃない(笑)。

表　健一(18)石川県

「なぜかサザエさんが見たくてたまらなくなりました」というのはわかる気がするなあ。

◆私は仕事場への片道35kmぐらいを愛車のゴキブリCITY子ちゃんで通勤しているのですが，最近ある交差点の信号サイクルが変わったらしくて，それまでは大渋滞道路だったのがスイスイ走れるようになって喜んでいます。55分ぐらいかかっていたのが45分で到着できるようになりました。どうやら，香川県警では試行錯誤を繰り返しつつ信号サイクルを決めているようです。12月号の「シミュレーションプログラミング入門」を県警の人に読ませたいものだ。Oh!Xの読者の中には警察に勤めている人もいるのかなあ。ちなみに，私は検察事務官です。

長谷川　聖(26)香川県

そういう試行錯誤はどんどんやってほしいですね。なにしろ渋滞はイライラしますからね。

◆障害者のコミュニケーション手段として，32個のシンボルを表示し，ひとつのキー操作で並べ替えていくプログラムをX-BASICで作っています。チンタラやっているので完成はまだまだでしょう。……といっている間にPC-9801とMacintoshに包囲されてしまいました(島田療育園リハビリ部の中の話)。しかし，難波のシャープは単機でも生きています。絶対に。

深栖　嘉哉(34)東京都

最近こういう分野へのコンピュータの利用というのも盛んになってきているようですが，いいことですよね。

◆出そう出そうと思っていてアンケートハガキを2カ月分ポストに入れられなかった。年寄りには時間が立つのが早い。ますますC compiler PRO-68K ver.2.0がほしい！　しかし，私には購入できない。いま45,000円は出せないのです。この年になって免許のために自動車学校に行くなんて。ところで，古い話ですが娘の学校の文化祭に行ったら，昔はどこでもあった飛行機研究会や鉄道研究会がなくサビシイ。しかし，その代わりにパソコンクラブではMZ-1500を並べてゲームをしていた。　横山　紘一(45)埼玉県

鉄道研究会は結構まだありそうですけどね。飛行機研究会もチラホラとはあるんじゃないでしょうか。

◆つ，ついに私の住むアパートに地上げ屋が来た。幸いにして話し合いでケリがつき，来年3月までに出ていくことになった。他人事と思っていたことが我が身に起こるなんて思ってもみませんでした。Oh!Xの編集部の皆さんも風邪と地上げ屋に気をつけてお仕事がんばってください(そのうちにソフトバンクのビルも……)。

川田　剛(18)大阪府

ここにはまさか地上げ屋は来ないでしょうけどねえ。しかし，自宅のアパートとかに地上げ屋が来たら結構ビビルだろうなあ。

◆それにしても，X68000オリジナルグッズってのは一体なんなんでしょうね。キャラクターグッズなどというのはよく聞くけど，この場合単にX68000の文字が入っているだけじゃないですか。もっとも，ライターにはツタンカーメンの絵が彫られているが，古代エジプトのファラオのキャラクターグッズというのは聞いたことがないし，前13世紀に彼がアモン＝ラー信仰復活とテーベ復都を記念して自分のグッズを作ったとも思えない。それでもやっぱりほしくなってしまう。このようなものを作るなんて，目のつけどころがシャープだと思う。

坂本　康(18)秋田県

あれって結構人気あるんですよね。それほど皆さんX68000を愛しているということの表れでしょうか。

◆フロッピーディスクの磁性面を太陽に透かしてみると，かつて小学校の理科の実験で黒下敷きを使ってみたのと同じような効果があるではないか。しかも，メーカーによって色が違う(マクセルのは赤，コニカのはだいだいだった)。うーん，小学校が懐かしい。宗石　茂(17)宮崎県

大事なフロッピーディスクでやるのはなんとなく怖いから，今度いらないディスクでやってみようっと。

◆最近，X68000に電源を入れるのも面倒。ゲームソフトがハードディスクにインストールできるようになれば，この際無理をしてでもハードディスクを買いたいが。パソコンゲームはこういった点で手軽でない。起動時間が長い。プロテクトチェックでさらに長い。ゲーム中ディスクアクセスで待たされる。しかたないことだが。その点，ファミコンだとかはROMカートリッジを差し込んで即プレイできるし，パソコン本体のように大きくないのでコタツの中に入れてプレイできたりとかする。まあ，グラフィックとかはパソコンとは比較できないが。

三宅　宏典(18)岡山県

やっぱり手軽さという点ではファミコンなどのゲーム機にはかないませんよね。

◆文化祭でバンドをやった。もちろん，私はキーボード。オルガンやストリングスでコードを白玉で弾いていただけだが，いつもパソコンで機械的な音楽ばかり聴いている私にとって，人間的なノリはとてもいい刺激になりました。

P.S.メンバー紹介のときになにかパフォーマンスをやれというので，ゼビウスを弾いたらバカウケしました。　千葉　浩貴(18)宮城県

ほかの人と演奏するといろいろ勉強にもなりますし，いいことですよね。

◆ハードディスクってどのくらいの振動でクラッシュするんですかねえ？　私の家の前の道路は穴が空いていて，そのうえ制限速度20kmにもかかわらず大型トラックが60kmぐらいで通るので，震度1ぐらいの揺れがするんですよ。これじゃあ，買っても使えないかも(もっとも，メモリが先になると思うけど)。

高橋　定元(18)千葉県

ディスクを読みにいっているときとかに揺れがあると，やっぱりやばそうですよね。しかし，そんなこと考えているよりも先に道路の穴を直してもらったほうがいいのではないでしょうか。

◆我が家はインドアゲームを中心に毎晩毎晩わいわいがやがや。最近ではパズルに凝っています。パズルといっても範囲が広く一時はジグソーパズルに熱中していました。この頃，パズル雑誌(クロスワードを中心にしたもの)の創刊が続いています。いま，静かなブームでしょうか。

P.S. X68000でパズルを解くソフトを作っています(あんまり意味ないか)。

松下　博(52)愛知県

適度に頭を使うというのはいいことだし，なんといっても気持ちがいいですもんね。あんまり難しいものになるとイライラしちゃうんですけどね。

◆「JR東日本がスキー場を作りました」とか。民営化するとこんなにも変わってしまうのかと驚きました。民営化したために我が家のすぐ近くの新駅設置の話が遠いところへいってしまったというのに。　瀧田　智康(22)埼玉県

こんなところにしわ寄せが……。

◆いつも1日では読み切れないような内容のOh!Xの編集部の皆様，こんにちは。本屋へ行くとよく思うんですが，このOh!Xに連載された(されている)記事が単行本になっていないのはなぜですか？　違う雑誌のやつはよく見かけるのに……。ぜひ出してください。そういうわけでこれからも“マニアック”な内容のOh!Xにしてください。　大栗　登(17)愛知県

連載が単行本になっているものもありますよ。最近では，「X68000マシン語プログラミング」が出たので，どうぞよろしく。

◆1年前にX68000を買うまでは，同じ県内にある矢板の工場を意識することはなかった。そもそも，シャープという会社自体を二流としか見ていなかった。しかし，いまでは矢板の工場をソンケーの眼差しで見つめ，「X68000は栃木生まれのパソコンなんだぜ」と友人に自信を持って勧めています。SX-WINDOWのFirst-Class-Technorogyも宇都宮にあることを知り，ミョーな感慨に浸っています。　吉葉　勝幸(19)栃木県

心の持ちようで，同じものでも違って見えるものですよね。

◆私は消防署に勤務していますが，先日OA機器調査ということで，全国の消防署で使用されているパソコン機種とか，使用内容といった調査がありました。その中でパソコン機種一覧表というものがありました。該当する機種のところに丸印をつけるようになっていたのですが，なんとX68000の名前があるではあるまいか。はたして，全国の消防署の中でX68000を使用している消防署がどれだけあるのか？　あれはすばらしい！　消防署だから昼夜かまわず仕事しているふりをして遊べるではないか。うらやましい。私も職場でもX68000の顔を見たい。

内匠　勇二(31)和歌山県

確かに，使っているところが果たしてあるのかなという感じはしますよね。結果がわかれば教えてほしいですね。もちろん，その用途も。

◆全自動洗濯機を見ていて思ったのだが，いまの日本の科学力(?)をもってすれば，全自動炊飯器なんてのが出てもいいと思う。入れた米の重さと新米，古米かの区別を選んでおけば勝手に米を洗って指定の時間になると炊ける。そうそう，もちろんファジィで炊いてね。

伴　哲也(19)京都府

それはいい。米を洗うのは面倒臭いですからね。冬は水が冷たいし。洗うときはやっぱり回転するんだろうか。

◆11月21日，そうスーパーファミコンの発売日である。この日，私の友人であるR君は後輩が並ばずに買えたというスーパーファミコン(My first SONYのような箱に入っていた)のパッドのフタを開けて，回路図を引いたのだ。中には1本のジャンパー線も見られず，ミツミの内職のおばちゃんがリード線を挟みながらネジを締めたのがわかった(笑)。友人は「これでジョイスティックが作れる」といっていたが，L,Rボタンはたぶん前代未聞のフットスイッチになりそうである。本体はオーナーの許しが得られなかったので断念。しかし，それにしてもF-ZEROはハードの力をまざまざと見せつけてくれた。

小川　純一(17)埼玉県

並ばずに買えたとはなんと幸運な。まだ，本体を見たこともないというのに。まあ，別にほしくもないからいいもんね(負けおしみ)。

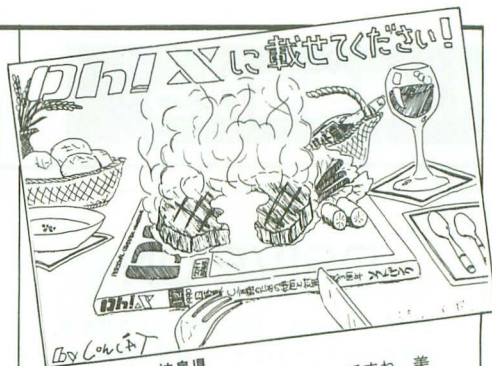
◆へっへっへ。やっと金がたまったぜ。これで1Mバイト増設RAMが手に入る……，と思ったんだけれど今日はスーパーファミコンの発売日ではないか。すまん，X68000よ！　もう半年待ってくれ。　福永　啓二(17)広島県

こういう具合に，スーパーファミコンの話題というのも結構多かったのですが，12月号のハガキで多かったのは“やはり”この

◀杉浦　豊　大阪府
本当に季節外れですが，ギャラガ'88のイラストということですから。冬は夏が恋しくなり，夏は冬が恋しくなる。人間ってわがままだなあ。

◀味野　真一　岡山県
これもちょっと載せるタイミングがずれてしまいましたが，ほのぼのしていていいでしょう。サンタクロースも大変だなあ。

話題です。
◆12月号を読むとカラーページにグラディウスのようなゲームの画面写真が載っている。X68000のネメシス'90かと思い，よく見るとそれはグラディウス。正方形のような画面からFM TOWNS版グラディウスかと思ったが，記事に目を移すとX1turbo専用と書かれている。しかも，デジタル8色で動きはまともだというからすごい。やはり，X1turboに不可能はない。きっとそのうちアフターバーナーも移植されることでしょう。　森山　伸行(19)東京都
◆ぬうう。「THE USER'S WORKS」。最初に見たとき「なんでいまさらX68000のグラディウスを？」などと思った。スゴイ。ホントにこれこそ真の"USER'S WORK"だ。シャープのパソコンって誰よりもユーザーに愛されているんだなあ。X1やMZシリーズって幸せかも。ところで，1989年10月号の原さん，X1turbo Z専用シューティングは？　高橋　明(20)茨城県
◆「THE USER'S WORKS」を見たとき，感動のあまり声も出ずふるえあがってしまった。いろいろな問題を抱えているが，ぜひ世に送り出してほしい作品であった。本当にX1turboには不可能はないのであろうかと思わせるものだ（グラフィックだけでも)。いま，衰えてきているX1の灯を再びともすきっかけともなってほしい。
松尾　保孝(18)奈良県
◆X1turboでグラディウス，それも超美麗なグラフィック。執着ともいえるほどまでのこだわり，オプション12個でも平気なプログラミングテクを抱えて登場した。ぜひとも完成させてほしい。このような作品を見ていると，まだまだX1も捨てがたいですね。なんでもできそうな気がします。私も大学にはいれたらなにか作ろうと思います。改良版X-700もどきを使って……。ふっふっふ。　荻野　友隆(17)京都府
◆す，すごすぎる……。同人ソフトなんて興味ないもんね，なんて読み飛ばそうとしていたら，「アレ」ですものね。「アマチュアの鑑！　よっ，大将ニクイねっ，この！」という感じ(?)です。とにかく作者の横内君には「がんばって完成してください」，そういってあげたいと思う22歳の大学生なのでした。　斎藤　修(22)宮城県
◆12月号をぺらぺらとめくって読んでいたら，いきなりグラディウスの写真が載っていた。ゲーム特集なのかなあとよく読んでみると，アマチュアプログラマが作ったX1turbo用のグラディウスであった。本当にあのデジタル8色のX1turboで作ったのかと疑うくらいだ。実際に動いているところやBGMを聞いていないからわからないが，機会があればぜひプレイしてみたいと思う。X68000のグラディウスと見まちがえてしまった。ノーマルX1でもできたらなあ……，でもすごい。　大田　哲矢(18)神奈川県

ここに載せたのはほんの一部で，本当にすごい反響でした。まあ，あの出来を見れば驚いたり感動したりするのは，当然のことといえば当然なんですが。横内君もこれを読んで，またプログラミングに励むことでしょう。

▲上田　修　岐阜県
イラストでこういう題材はめずらしいですね。美術の時間とかを思い出しそうです。果物とか瓶とかはよく描かされましたよね。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲　間

★X68000でフォントの研究開発を行っている「タイプ・ラボ」では会員とスタッフを募集しています。日本語版組システムTeXなどを中心に活動しながら、不定期でディスク会報も発行しています。62円切手同封のうえ、下記まで。〒910 福井県福井市渡町358-4 平木敬太郎(23)
★「NANNO CLUB PRO-68X」では、X68000ユーザーを対象に南野陽子ファンを募集。活動はSX-WINDOWのアプリケーションの配布、パソコン通信でのリアルタイム情報提供など。南野陽子情報日本一宣言中です。申し込みは62円切手同封のうえ封書で。〒332　埼玉県川口市川口6-5-40中銀マンション605　武山智裕(18)

売ります

★CZ-8PK9(24ピン80桁，付属品，箱あり)を3万5千円。X1turboテクニカルマニュアル(シャープ監修　AZビシコム刊)を5千円。PC-E200(ポケコン，Z80CPU，32Kバイト)を5千円。いずれも送料込み。連絡は往復ハガキでお願いします。〒183　東京都府中市多磨町1-34-7（株)ジャコム多磨寮　山口幸一(24)

買います

★シャープカラーイメージユニット「IO-735X」を8万円ぐらいで。安い人優先です。オプション（X68000用ケーブル等)をつけてくれる方は多少値上げします。連絡は往復ハガキで。〒981-31　宮城県仙台市泉区長命ヶ丘6-6-12　石川太朗(15)
★熱転写カラー漢字プリンタ「CZ-8PC3」を2万円以内で(安価優先)。RS-232C，マウスボード「CZ-8BM2」＋マウス「CZ-8NM2」を7千円前後で。連絡は状態，付属品の有無，電話番号を明記のうえ，往復ハガキで。〒319-01　茨城県新治郡八郷町東成井1690　菱沼義博(16)
★X68000用数値演算プロセッサボード「CZ-6BP1」を3万円前後で。プリンタ「CZ-8PC4」4万5千円前後で。どちらも取扱説明書，付属品ありのものを。連絡は希望価格を書いてハガキで。〒731-42　広島県安芸郡熊野町柿追49　長石裕行(21)
★X1用フロッピーディスクドライブ「CZ-503F」か「CZ-502F」を3万円以内で。漢字ROM「CZ-8BK2」を5千円前後で。また，X1用のプリンタを1万円から1万5千円で。完動品なら可。どれも送料込み。連絡はハガキで。〒065　北海道札幌市東区北11条東4丁目　荒井真公人(18)
★カラーイメージユニット「CZ-6VT1」を2万5千円で。アスキーのテクニカルデータブックを定価以上で。連絡はハガキで。〒923　石川県小松市飴屋町21　紙山満(17)

バックナンバー

★Oh!X1990年1、3、5月号広告などの切り抜き可。定価で譲ってくださる方。送料はこちらで持ちますのでよろしく。連絡はハガキで。〒813　福岡県福岡市東区名島3-29-16　仁泉大輔(19)
★Oh!Xの1989年6月号を送料込み1,500円で買います。切り抜き不可。少々の汚れは可。連絡はハガキで。〒729-24　広島県豊田郡安芸津町風早649-366　梶岡真二(17)
★Oh!MZの1985年1、4、6～9月号を送料込み各千円で。切り抜き不可。連絡は往復ハガキで。気長に待ちます。〒611　京都府宇治市宇治蔭山55番地331A　榊卓也(24)
★Oh!MZ1987年5、6月号，Oh!X1988年7月号を送料込み各1,500円で。切り抜き不可。連絡は往復ハガキで。〒123　東京都足立区江北7-12-1-506　村松秀雄(37)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は12月号の内容に関するレポートです。

●「アナログジョイスティックの製作」は参考程度に，ということで割り切って読ませていただきました。

回路の規模も結構大きなものですし，普通Z80を使うならシングルボードコンピュータを買ってきますしね。また，ROMライタが必要であることで，多くの人にとって非現実の世界になってしまったでしょうし。

あくまで「体験記」であり，製作してもらうことが目的ではないという意味では大成功だと思います。リストだってダンプではなくソースで（ついでにZEDA用）であることもその主旨に沿ったものであったといえるでしょう。 時間の都合で私はまだこのようなZ80を使ったコントローラを作ったことはありませんが，いずれ作るときには大変参考になると思います。

アナログスティックは，あのメカ部がネックになっていままで製作できなかった人もいたと思います。私もラジコンスティックの回路をアレンジして，完全ハード版サイバースティックを考え，回路図まではできていたんですが，例のボリューム式のジョイスティックが手に入らなかったことで計画を断念しました。Apple IIやVIC-1001の時代には，日本橋にもパドル用にと売っていたのですが，いまはさすがになし。やむをえず親しいパーツ屋の人に「メーカーさんに分けてもらってください」とお願いする始末。入手しにくいパーツであることは間違いないようです。

サイバースティックは23,800円でも，XE-1APなら1万円程度で買うことができます。いさぎよく完成品を買うほうが賢いんでしょうね。どちらにしても，こういったトランジスタ技術にしか掲載されないような記事がこのOh!Xに載ったことは，Oh!Xのキャラクターが反映されていて安心できました。

浅野 憲（19） X1turboIII，X1F model 20，FM-77L2，M5Jr. 大阪府

●特集「XCのための傾向と対策」について。記事のレベルはいまくらいで適当だと思います。本気で始めようと思ったら，たいていなにかしらの本を購入するでしょうし，いまくらいの記事ならC言語に入門しようかと迷っている人を誘い込むのには適度な難易度だと思います。あと，毎年1回はこの関係の記事があるのですから，半年ごと（または1年ごと）の課題を作ってプログラムを募集するのもいいのではないかと思います。

中川 比呂志（19） X68000，X1cs 東京都

●新連載「シミュレーションプログラミング入門」について。冒頭で月産台数が3ケタも出るかどうか，あやしいような車が出てきたのには思わず笑ってしまったが，車の交通をテーマに持ってきたのは正解だと思う。日頃，交通渋滞を経験せずにすむ人間でも，赤信号で止まっていらいらしながら待つことは，ごく一部の例外（離島で信号がないとか，そもそも車がないとか）を除いて誰もが経験しているのだから，関心を持たないわけがない。また，ほかの交通機関（たとえば鉄道）のように特別なルールを知っている必要性もなくて，ごくごく日常的な知識の範囲で理解できるのだから，これはすごくいいテーマ選びだと思う。

「アナログジョイスティックの製作」について。ROMライタがないとお話にならないということを別にして考えると，12月号のこの記事はいろいろと考えさせられることがあって，面白かったと思う。なにより「Z80はRAMなしでも使える」というのが気に入った。RAMと一緒になって使われているものしか見たことがなかったせいもあるが，「CPUとはメモリなしでは使えない」という固定概念にとらわれてしまっていた自分が妙に情けなくなってしまった。

土谷 興正（19） X1G，MSX2 兵庫県

●C言語について。実際にCをフルセットでガンガン使ったことはないのでたいしたことはいえないが，Small-Cを使ったかぎりではマクロがいっぱい使えるアセンブラみたいで便利だと思う。ただ，一応標準規格とも呼べるANSIのCが登場した以上は，過去のしがらみを捨ててフルコンパチ（アッパーコンパチは不可）にしないと，BASICのようにグチャグチャになっていってしまう可能性が高いと思う。

特に，MS-CとQUICK-CとTURBO-Cの3者は足並みを揃えるべきだと感じるが，いまのところはX68000には関係ないですね。

高村 信（20） X1turbo model 30，PC-8001mkII 東京都

●「アナログジョイスティックの製作」について。ハードについてはわかりませんが，体験したことをそのまま書き連ねていくことによって，読者のほうにしてみれば実際に作る作らないは別として作ったような気分，あるいは作りたい気分になるのではないのでしょうか。

安井 百合江（16） X68000 PRO 愛知県

●X1turbo用「グラディウス」について。X1turboユーザーにとり，いい刺激になったのではないでしょうか。MZ-700ユーザーが「スペースハリアー」を見て奮起したように，きっとこれでX1turboユーザーも「X1もまだまだ」と奮起し，昔作りかけで挫折してしまったプログラムを引きずり出してくる気が起きたのではないでしょうか。諸々の問題はありましょうが，X1turboユーザーの起爆剤となったであろう，この作品を載せたことは正解であったと思います。

高橋 毅（19） X68000 PRO，MSX2 埼玉県

ごめんなさいのコーナー

1月号 Z's-EX

Z'sSTAFF ver.1.0ではZ's-EXで拡張されたウィンドウの表示がおかしくなるなどの症状が出ています。原因は画面の初期化関係に違いがあるためです。基本的にはバージョンアップサービスを受けることをお勧めしますが，ver.1.0で利用したい場合には，Zs_EX.Xを起動する前に，

SCREEN 1，3，1

を実行して画面を初期化してください。

1月号 DICTOOL

ディスクには開発途中バージョンが入ってしまいました。本誌に掲載された完成バージョンは4月号の付録ディスクにあらためて収録する予定です。申し訳ありませんでした。

12月号 （で）のショートプロぱーてぃ

とろける.Xの作者名が間違っていました。正しくは市岡孝浩さんです。大変ご迷惑をかけました。お詫びいたします。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

X68000ならではのグラフィックパワーを追求しよう

▼厳しい年末進行のなか，お届けしたグラフィック特集ですがいかがでしたでしょうか。大きな表現力を持つX68000ですが，それだけに消費する資源もヘビーです。また，それを扱う人間の能力，体力，時間も大変なもの。今回は丹明彦氏の大活躍で切り抜けたものの，編集部ではグラフィック関係を担当してくれるスタッフを必要としています。力を貸してくれる方，ぜひご連絡ください。

▼1月号に引き続いて特集したSX-WINDOWですが，プログラミングの解説記事は1月号の付録ディスクに収録した資料をもとに行ってます。ディスクには各マネージャ（GRAPH MANなど）の働きを記したドキュメントファイル，SXコール利用のためのリファレンス，またC言語で必要となるライブラリおよびインクルードファイルなどが収められています。これらは今後ともウィンドウプログラミングの基礎資料として必要ですので，ぜひとも常備しておいてください。1月号をお持ちでない方は最寄りの書店でお申し込みください。なお，バックナンバーの在庫は1年間を目安に確保しておりますが，号によっては早く品切れになってしまう場合もありますのでご了承ください。

▼村田敏幸さんの「X68000マシン語プログラミング」が単行本化されましたが，好評につき品切れとなる書店が続出し，まだお求めになれない方も多いようです（ごめんなさい）。すでに重版がかかっておりますので，お待ちの方もまもなく入手できることと思います。また，お読みになったご意見ご感想などもお寄せください。

▼今月の「PASCALプログラミングへの招待」はお休みさせていただきます。また，「清水和人のプログラミング道場」は清水和人氏がドイツに滞在中のため連載が中断しています。春には帰国の予定ですのでしばらくお待ちください。なお，予告にありましたZ-BASIC用「サザエさん劇中曲」は版権の問題で掲載できませんでした。お詫びいたします。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋪㋮㋯」係

SHIFT・BREAK

▶最近，ドラゴンセイバーが横浜のゲームセンターに入った。と，そこにはドラスピ，フェリオスでクリアを競ったヤツがいた。前回，2回ともそいつに負けているので，「今回こそは勝ってやる」と，意気込んでプレイしているが，現在，彼は最終面そして私は6面でハマっている。う〜ん，どうやら勝負は見えているようだな。（純）

▶年末進行のおかげで，徹夜で原稿を書くことになった。そうと決めてしまうと，なんかそれだけで物事が解決したような気分になってしまって，ちっとも進まない。しかも，ここは誘惑材料にはことかかないし。おまけに，徹夜ボケの原稿は怖くて自分でも読めない。かくしてもう一晩徹夜するはめに。とほほ。（浦）

▶この本は2月号だな。「私の誕生日は2月です。さて何日でしょう？」という問題を小学生以来出し続けているが，いままで一発で当てられた確率は7割を越えている。特に女の子の正当率が高い。そして，そのあとの言葉も5割がた同じである。「どっちのプレゼントかわからないよね〜」う〜ん，今年も冬がやってきた。（亀）

▶車で編集室に通うと，終電の心配をする必要がないので気が楽になる。おかげで帰りが明け方になることも多くて，がらがらの道をカッ飛ばして帰るのが癖になってしまった。近ごろ自分でも運転マナーが悪くなったと感じている。免許も更新することだし，安全運転を志していこうと思う。（半年で14,000km走ったダークグリーンシルビアのH.K.）

▶この1年を振り返ってみると，海外のゲーム（の移植作品）にばかり熱中していたように思うのだ。こんなことではいけないのだが，面白いものは面白い。最近は海外ソフトの移植が流行のようだが，ただ上っ面の移植だけでなく，その底に流れるセンスをつかまえ，自分のものにしていきたい。とりあえずは腐った移植が山ほど出ないことを祈ろう。（A.T.）

▶本文中では「雪不足」と書いたが，どうやらようやく雪が降ってきたそうだ。降るほうに予想がはずれるぶんには大歓迎。今日，仕事を終え「イッパ」梅池に向かう。ところで，ついに「ニューロ」家電なるモノが誕生。このぶんだと次は「バイオ」かな。でもバイオの次にはいったい何がくるのだろう。皆さんはどう思われますか。（C）

▶にもかかわらず，何年か振りにTVドラマを見た。クリスマス・イブ。複雑な男女関係が，三角関係どころか四角，五角と増えていき，どんどん泥沼へはまっていくすごい話だ。私は思った。このまま六角，七角，と増えていけばn角のnが十分大きいと円に近似できるのと同様，すべてが丸く治まるのではないだろうか，とね。誰か，実験してみません？（K）

▶スーパーファミコンをなぜか買うことができた。が，箱を開けてびっくり。ACアダプタやRFケーブルが入っていない。ファミコンのものが代用できるからいいものの，画面のすごさを売りにするならビデオケーブルくらいは付属してほしかった。まあ，マリオ程度のソフト（絵が汚いぞ）しかできないのならそれでもいいか。（少し後悔してるKO）

▶1990年もなんとか無事に終わり（そうで），とりあえずはひと安心。就職という転機の年だったわけだが，なんとなく日々に追われ，あっという間に過ぎ去ってしまった。生まれてこのかた，ボーッとしながらのんびりと1年を過ごしていたので，こんなに早く1年が過ぎることはなかったのに。しかし，とりあえず充実した年ではあったなあ。（A）

▶ドアの外でガヤガヤと話し声。なんだろ，と思ってたら，あろうことか突然人ん家のドアを掃除し始めた！　な，なんだあ？　「あのぉ」「あ，すいませんねぇ，連絡しないで」。なんと，同じ階段の主婦連が勢揃い。なんでこの人たちが階段掃除してんのよ！　月1万円近くの管理費はどこいってんの？（それでも手伝えなくってゴメンナサイと思うE.O.）

▶同人誌を買うにも合い言葉が必要な今日この頃。POPULOUSはマニュアルを見なくても操作できたのにPOWER MONGERはいまいちわかりにくい。動きは軽い。もっと重たいゲームだと思い込んでいたのだが。しかし，膨大な効果音とグラフィック，これでディスク1枚組，要512KバイトRAM……イギリス人恐るべし（U）

▶24日のクリスマスイブは編集部で徹夜だった。そのまんま25日も徹夜になりそうだったが，かろうじて夕方7時ごろに家に戻って睡眠をとった。そして終電で出勤。クリスマス音楽のカセットをかけながら編集部に向かった。ああ，それなのに，会社のあるビルの玄関に待っていたのは，大きな憎らしい門松であった。ううっ……。（正月嫌いのT）

microOdyssey

ふと目が覚めると，約束の時間はとうに過ぎていた。げげーっ，やばい！ 指が憶えたTelephone numberを急いでなぞる。「はい，Oh!Xです」「あのぉ～，寝坊しました。すみません」「そんなこったろうと思った」，あらま……。

受話器を置いたら，なんかホッとした。時計はAM7：38を指している。ということは，2晩徹夜していたとはいえ7時間も眠ってたのか！ ここんとこ忙しかったし，いろいろあったからなぁ。あ，レーザーディスクひと晩回しっぱなし！ ま，いいか。このまま見ちゃおうっと。

雲が，流れてく，早く，早く……。……まるでいまのあたしみたい。流されるまま流されて，何処へたどり着くのかさえわからない。……マズイな，なんかnervousになってる。nervous breakdownてヤツかな。きっとそうだね。だって，1990年があたしにとって長かったのか短かったのかすら，もうわからないんだもの。いろいろなことがあったのは確かだけど。

でも，ほんとにこの1年間，自分自身のためになにかしたことってあっただろうか。

空が，青い，文字どおりの，Sky blue……。こんなきれいな空，最後に見たのはいつだったろう。見上げればいつでもあったはずなのに。灰色に見えたのは，あたしの心のせい？ モニタに映し出された空は，子供の頃に見ていた色とおんなじで，ほんとに青くて，ただそれだけなんだけど，涙が出そう。あ，出ちゃった。

やさしくて力強い，空の色から伝わってくるそんな思いが，胸に刺さってずきずきする。

モニタには，人波に逆らってこちらに向かってくる彼ら。それはまるで彼らの生きざまを表しているかのよう。流れに逆らってでも自分らしく生きる。つらくないわけがない。それでも彼らは笑っている。なぜかはあたしが，あたし自身の身体が，いちばんよく知ってる……。

そばにあるキーボードの鍵盤に，そっと触れる。なんだかとても懐かしい気がする。ギターを手に取る。指に触れる弦がちょっぴり痛いや。子供の頃から音楽が好きだった。Melodyさえあれば幸せ，なければ自分で唄っていた。気に入らなきゃ自分で作って。音楽をもっと表現したくてやったMusical，自分のPromotion videoに使いたくて勉強したCG。一生懸命だった頃。

ほんとに好きなことがあれば，なんだってやれる。この一瞬は二度と来ない，だから死んでるように生きたくないって，そう思ってたはずなのに。いつのまにか流れに逆らうことをやめ，濁流の中でもがいている。なんか滑稽だね。

いま，やっと見えてきた，ほんとうにやりたいこと。あたしにしかできないこと。

人を愛して別れていくなかで，あたしは時間を止める方法を教わった。もう25だから，そんなこと考えていたら，一歩も前に進めやしない。傷つくことも多いけど，うれしいこともたくさんある。いつでも一生懸命でいたい。たったひとつしかない，宇宙から見たらほんとにちっぽけだけど，あたしのmicroOdyssey……。

ぎゃ～，もうこんな時間！ 早く会社いかなくっちゃ。っと，レーザーディスクは消して。ほうら，外には真っ青な空が広がっているよ。

(12/24，ひとりのX'mas eveに。E.O.)

＊LASERDISC is "TWO ALONE UP-BEAT" by VICTOR MUSICAL INDUSTRIES.INC.

1991年3月号2月18日(月)発売

特集 MIDI & MUSIC PROCESSING
MUSICDRVの活用
MIDI楽器制御の基礎知識

SX-WINDOWプログラミング

X68000CARDDRV用カードゲーム EIGHT

Oh! X LIVE in '91
X68000用 戦いの兜
X1用 BIRTHDAY

バックナンバー常備店

地域	地区	店名・電話
東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートB1 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3200)9185
	渋谷	大盛堂書店 03(3463)0511
	池袋	リブロ池袋店 03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 2月号
■1991年2月1日発行 定価560円(本体544円)
■発行人 孫 正義
■編集人 橋本五郎
■発売元 ソフトバンク株式会社
■出版事業部 〒108 東京都港区高輪2-19-13 NS高輪ビル
Oh!X編集部 ☎03(5488)1309
出版営業部 ☎03(5488)1360 FAX 03(5488)1364
広告センター ☎03(3297)0181
■印 刷 凸版印刷株式会社

©1991 SOFTBANK CORP. 雑誌 02179-2本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。

バックナンバー案内

ここには1990年2月号から1991年1月号までをご紹介しました。現在1989年12，1990年9～12，1991年1月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については，188ページを参照してください。

1990

2月号（品切れ）
特集　画像圧縮へのアプローチ
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
●X68000用ゲームプログラムGonGon
●MZ-700用紙芝居Eyelarth
LIVE in '90　オーダイン/魔女の宅急便
THE SOFTOUCH　A-JAX/フラッピー2/夢幻戦士ヴァリスII
マジックパレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム　超小型コンパイラTTC++

3月号（品切れ）
特集　MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
●X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90　パワードリフト/スキーム/となりのトトロ
THE SOFTOUCH　ナイトアームズ/斬/ダンジョンマスター
全機種共通システム　超多機能アセンブラOHM-Z80

4月号（品切れ）
特集　ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040，ついに登場
LIVE in '90　バーニングフォース(OPMD対応)
THE SOFTOUCH　The Fille Professor/HOST PRO-68K
全機種共通システム　ファジィコンピュータシミュレータI-MY

5月号（品切れ）
特集　BASICプログラミング
第5回　言わせてくれなくちゃだワ
連載　ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90　TURBO OUTRUN
THE SOFTOUCH　天下統一/ポピュラス/Hyperword
全機種共通システム　インタプリタ言語STACK

6月号（品切れ）
特集　創刊8周年記念PRO-68K(付録5"2HD)
Oh!Xアンケート結果大分析大会
連載　ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90　ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH　三国志II/FAR SIDE MOON/グラナダ
全機種共通システム　X68000用S-OS"SWORD"他

7月号（品切れ）
特集　マシン語への第一歩
X68000SUPER-HD試用レポート
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1――ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90　夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH　サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム　リロケータブルアセンブラWZD

8月号（品切れ）
特集　ADVANCED 2D GRAPHICS
100号記念特別モニタプレゼント
連載　ショートプロぱーてぃ/Z80's Bar/INTEGRAL X1
X-BASIC調理実習/X68000マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●X68000用画像回転プログラム　XROT0.X
LIVE in '90　OMENS OF LOVE/ENDLESS RAIN/ダートフォックス
THE SOFTOUCH　大航海時代/ウルティマV/プロミストランド
全機種共通システム　リンカWLK

9月号
特集1　日本語を処理するための序章
特集2　ADVANCED 2D GRAPHICS
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90　風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH　T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム　BILLIARDS

10月号
特集　電子音楽術入門
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語
LIVE in '90　Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH　ワールドコート/ルーンワース/闇の血族/提督の決断
全機種共通システム　ライブラリアンWLB

11月号
特集　理科系のGAME REVIEW
連載　Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL X1
荻窪圭の大人のためのX68000
LIVE in '90　ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL　ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム　スクリーンエディタEDC-T

12月号
特集　XCのための傾向と対策
連載　X-BASICプログラミング調理実習/ハードウェア工作入門
マシン語プログラミング/ショートプロぱーてぃ/Z80's Bar
大人のためのX68000/ようこそここへC言語/INTEGRAL X1
●シミュレーションプログラミング入門
●特別企画アナログジョイスティックの製作
LIVE in '90　グラディウスIII/メタルサイト
THE SOFTOUCH SPECIAL　イメージファイト/ジェミニウイング/NAIOUS他
全機種共通システム　STACKコンパイラ

1991

1月号
特集　急接近！ SX-WINDOW
特別付録　謹賀新年PRO-68K(5" 2HD)
連載　ハードウェア工作入門/シミュレーションプログラミング入門
DōGA・CGA/ショートプロぱーてぃ/大人のためのX68000
PurePASCAL/清水和人流プログラミング道場/X-BASIC調理実習
LIVE in '91　めぞん一刻/涙で綴るパパへの手紙
THE SOFTOUCH　ソル・フィース/銀英伝II/続ダンジョン・マスター他
製品紹介　光磁気ディスクCZ-6MO1
全機種共通システム　ブロックアクションゲームCOLUMNS

ソフトバンクの書籍特約書店

下記の書店の一覧は、ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。

(注)現品が売れて補充中の場合もございますので、ご注意下さい。

SOFT BANK

ソフトバンク出版事業部

〒108 東京都港区高輪2-19-13 ☎03(5488)1360

全国特約書店一覧

地域	書店名	電話番号
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善らがぁーる新札幌DUO店	011-890-2588
〃	富貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	今泉本店	0172-32-2231
〃	メディアイン城東店	0172-27-8118
八戸市	伊吉書院	0178-44-1917
盛岡市	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	髙山書店	022-263-1511
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
〈関東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカヰ書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	048-822-5321
〃	須原屋コルソ店	048-824-5321
大宮市	押田謙文堂	048-641-3141
〃	ブックセンター押田	048-647-3141
〃	三省堂ブックポート	048-646-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	048-752-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	厳翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二厳翠堂	0474-65-0926
〃	三省堂書店西船橋店	0474-34-3111
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東ロルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	文教堂溝の口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本店	0427-74-5581
相模原市	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
〈東京〉		
千代田区	三省堂書店神田本店	03-3233-3312
〃	書泉グランデ	03-3295-0011
〃	東京堂書店	03-3291-5181
〃	旭屋書店水道橋店	03-3294-3781
〃	丸善お茶の水店	03-3295-5581
〃	厳翠堂	03-3291-1362
〃	いずみ神田南口店	03-3254-8521
〃	明正堂秋葉原店	03-3257-0758
〃	Bit INN 東京	03-3255-4575
〃	T-ZONE	03-3257-2660
〃	ラオックス THE COMPUTER館	03-5256-3111
中央区	八重洲ブックセンター	03-3281-1811
〃	日本橋丸善	03-3272-7211
〃	旭屋書店銀座店	03-3573-4936
港区	書原新橋店	03-3591-8738
〃	雄峰堂NS店	03-3503-6586
〃	虎ノ門書房本店	03-3502-3461
〃	虎ノ門書房田町店	03-3454-2571
品川区	芳林堂大井町店	03-3474-4946
〃	明屋書店五反田店	03-3492-3881
渋谷区	紀伊國屋書店渋谷店	03-3463-3241
〃	旭屋書店渋谷店	03-3476-3971
〃	三省堂書店渋谷店	03-3407-4545
〃	大盛堂書店	03-3463-0511
〃	紀伊國屋書店笹塚店	03-3485-0131
新宿区	紀伊國屋書店本店	03-3354-0131
〃	三省堂書店新宿西口店	03-3343-4871
〃	福家書店センタービル店	03-3345-1246
〃	福家書店野村ビル店	03-3342-0298
〃	新星堂NSビル店	03-3344-2055
〃	西武新宿ブックセンター	03-3208-0380
〃	芳林堂高田馬場店	03-3208-0241
〃	未来堂	03-3200-9185
豊島区	旭屋書店池袋店	03-3986-0311
〃	芳林堂池袋店	03-3984-1101
〃	リブロ池袋店	03-3981-0111
〃	三省堂書店池袋店	03-3987-0511
〃	新栄堂本店	03-3984-2345
〃	新栄堂アルパ店	03-3988-0181
台東区	明正堂中通り店	03-3831-0191
墨田区	ブックストア・談	03-3635-1841
葛飾区	文教堂青戸店	03-3838-5938
江戸川区	文教堂西葛西店	03-3689-3621
大田区	アクトブックスサンカマタ店	03-3735-1551
〃	竜文堂大森駅ビル店	03-3775-3851
中野区	明屋書店東京本社	03-3387-8451
杉並区	ブックセンター荻窪	03-3393-5571
〃	書原杉並店	03-3313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211

特約書店基本図書一覧

定価はすべて税込です。

分類	書名	定価
アセンブラ	8086アセンブリ言語	●2,890円
アセンブラ	8086マクロプログラミング	●2,680円
アセンブラ	入門Turbo PASCAL Ver.5プログラミング	●3,300円
アセンブラ	GDCテクニカルブック	●3,500円
C言語	C言語の基礎知識	●2,580円
C言語	C言語の活用理解	●2,060円
C言語	C言語の応用50例	●2,370円
C言語	上級・C言語の応用50例	●2,480円
C言語	Play the C 上巻	●1,550円
C言語	Play the C 下巻	●1,550円
C言語	Cプリプロセッサ・パワー	●2,270円
C言語	Turbo C 入門	●2,680円
C言語	C++プログラミング	●2,680円
C言語	Quick Cプログラミング	●2,680円
C言語	詳説C言語	●4,500円
C言語	MS-C Ver.5.1プログラミング	●3,300円
C言語	Turbo C Ver.2.0プログラミング	●2,900円
機種別	ダイナブック・スーパーガイド	●3,200円
機種別	最新ハードディスク入門	●2,600円
機種別	最新EMS・RAMディスク入門	●2,500円
OS	プレイMS-DOS	●1,960円
OS	MS-DOSいたれりつくせり本	●1,860円
OS	新MS-DOS入門 ビギナー編	●1,900円
OS	新MS-DOS入門 シニア編	●2,300円
OS	新MS-DOS入門 応用編	●2,300円
OS	OS/2 APIブックI	●2,790円
OS	OS/2 APIブックII	●3,000円
OS	UNIXオペレーティング・ガイド	●3,090円
ワープロ	一太郎 Ver.3 ガイド	●2,580円
ワープロ	入門一太郎 Ver.4.2	●2,500円
ワープロ	P1 EXEガイド	●2,600円
ゲーム	RPG幻想事典	●1,550円
ゲーム	RPG幻想事典 日本編	●1,860円
ゲーム	魔法王国シムルグント	●1,860円
アプリケーション	LOTUS1-2-3ガイド ビギナー編	●2,480円
アプリケーション	LOTUS1-2-3 ガイドII	●2,580円
アプリケーション	桐Ver.2ガイド	●2,580円
アプリケーション	入門桐 Ver.2 一括処理	●3,500円
アプリケーション	Ninja 3ガイド	●2,300円
アプリケーション	MS-Chart Ver.3.1ガイド	●2,990円
アプリケーション	まいとーくガイド	●2,370円
アプリケーション	d BASE III PLUS ガイド	●3,800円
アプリケーション	The CARD 3 ガイド	●2,900円
情報処理試験	アセンブラCASL入門	●2,060円
情報処理試験	ハードウェア徹底マスター	●2,580円
情報処理試験	FORTRAN徹底マスター	●2,890円
情報処理試験	受験用語ハンドブック	●1,860円
情報処理試験	情報処理入門 I 基礎知識	●1,240円
情報処理試験	情報処理入門 II 関連知識	●1,240円
情報処理試験	CASLで学ぶアセンブラ入門	●2,270円
情報処理試験	そっくり模擬試験	●2,200円

所在地	書店名	電話番号
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
〃	アクロスブックセンター	0263-32-5733
上田市	平安堂上田店	0268-22-4545
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
上越市	ハソトヒア コスモス	0255-25-5867
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東 海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店S B·S店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルハーク	052-971-1231
〃	日進堂上前津店	052-263-0550
名古屋市	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
〃	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
豊川市	三洋堂豊川店	05338-3-0334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
一宮市	三洋堂一宮店	0586-77-5734
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイハス店	0584-74-7766
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店IIビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近 畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-624!
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中 国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
福山市	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
〃	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四 国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	092-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館佐賀店	0952-24-4314
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	096-322-5531
〃	長崎書店	096-353-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

満開の電(でん)子ちゃん

作：いわいいっぺい
え：岡村 祭

brother
パソコンソフト自販機
ソフトベンダー武尊

札幌	そごう電器YES 5F	(011)214-2850
	パソコンショップハドソン	(011)205-1590
青森	電巧堂チェーン青森本店	(0177)23-2356
盛岡	デンコードー盛岡本店	(0196)54-2772
秋田	電巧堂チェーン秋田駅前本店	(0188)34-3151
仙台	デンコードー仙台本店	(022)225-5290
	デンコードーDac仙台東口店	(022)291-4744
	庄子デンキコンピュータ中央店	(022)224-5591
郡山	うすい百貨店	(0249)32-0001
福島	庄子デンキ粉又駅前店	(0245)21-2011
いわき	いわきマイコンショップ	(0246)23-0513
山形	庄子デンキ山形七日町本店	(0236)42-1222
新潟	PIC	(025)243-5135
	PICこばり店	(025)233-5791
長岡	丸専デパート 1F	(0258)33-4970
	丸専マイコンショップジャスコ長岡店	(0258)27-6033
宇都宮	K&P宇都宮	(0286)62-0002
前橋	パソコンランド21 前橋店	(0272)21-2721
高崎	パソコンランド21 高崎店	(0273)26-5221
太田	パソコンランド21 太田店	(0276)45-0721
桐生	パソコンランド21 桐生店	(0277)45-2721
水戸	川又書店駅前店 2F	(0292)31-0102
つくば	ミドリ電気館	(0298)55-3715
大宮	ダイエー大宮店 6F	(0486)45-4147
川越	長崎屋川越新宿電気館 1F	(0492)44-5461
上尾	ポンベルタ上尾	(0487)73-8711
所沢	サン家電所沢店 2F	(0429)39-5117
松戸	イトーヨーカドー松戸店	(0473)68-5131
千葉	ラオックス千葉店(PERIE 3F)	(0472)27-5318
八千代台	ラオックス八千代台店	(0474)85-2261
秋葉原	丸善無線ECCS 4F	(03)3255-4911
	ミナミ電気館 4F	(03)3255-4040
	サトームセン(メディアセンター)	(03)5256-3267
	サトームセン(ラジオ館) 5F	(03)3251-1464
	サトームセン本店 5F	(03)3253-5879
	CVA秋葉原 2F	(03)3258-3711
	コム本店	(03)3251-1523
	ザ・コンピュータ館	(03)5256-3111
新宿	ラオックス新宿店	(03)3350-1241
池袋	ビックカメラ池袋東口本店 4F	(03)3988-0020
	ビックカメラ池袋北口店 4F	(03)3988-8666
渋谷	上新電機J&P渋谷店 1F	(03)3496-4141
	ビックカメラ渋谷店A館 6F	(03)3477-0002
世田谷	ヤマギワ玉川店(高島屋 5F)	(03)3708-1638
武蔵野	ラオックス吉祥寺店 2F	(0422)21-3471
小金井	サン家電小金井店	(0423)85-3810
立川	丸善無線立川店(Will 6F)	(0425)27-6211
八王子	ムラウチ電気 2F	(0426)42-6211
	J&P八王子店(そごう 7F)	(0426)26-4141
町田	東急ハンズ町田店 B1F	(0427)28-2603
	J&P町田店	(0427)23-1313
横浜	ソフトクリエイト横浜店	(045)314-4777
	横浜VIVRE 21 7F	(045)314-2121
	ダイエー戸塚店 3F	(045)881-1261
	ICコスモランドあざみの	(045)901-1901
	ICコスモランドかもい	(045)934-9636
川崎	セキグチデンキ館	(044)244-5421
藤沢	WAVE EYE	(0466)43-1771
平塚	梅屋	(0463)22-4147
厚木	ラオックス厚木店オーディオ館	(0462)22-2722
大和	WAVE EYE大和店	(0462)63-9898
甲府	中込電気商会	(0552)24-5431
長野	ダイエー長野店 7F	(0262)27-1311
松本	道兵	(0263)32-6350
金沢	うつのみや片町店	(0762)21-6136
富山	丸の内カラー駅前店	(0764)41-9075
福井	PASわくわくメディア館	(0776)23-7621
	だるまや西武 7F AVC	(0776)27-0111
沼津	メルバ沼津	(0559)22-4858
静岡	メルバ静岡	(0542)54-5338
富士	メルバ富士店	(0545)51-8022
浜松	メルバ浜松本店	(0534)64-8412
	ホーエー家電有楽店	(0534)53-1441
	マエダM&V	(0534)73-1691
名古屋	栄電社テクノ名古屋	(052)581-1241
	パソコンショップコムロード	(052)263-5828
	ちくさ正文館書店ターミナル店	(052)732-3601
	カトー無線電機電気館 4F	(052)264-1534
	丸善無線 第一アメ横店	(052)263-1626
	栄電社テクノ大須	(052)252-0181
	トップカメラ 4F	(052)971-0111
	J&P大須店	(052)262-1141
	エイデンメディア大須	(052)242-8591
豊橋	栄電社テクノ豊橋	(0532)52-1231
岡崎	ジャスコ岡崎店 3F	(0564)23-4960
豊田	ジャスコ豊田店パソコンショップJPC	(0565)35-1761
岐阜	パソコンショップコムロード岐阜店	(0582)66-0288
大垣	スイテック大垣ヤナゲン店A館 6F	(0584)81-3491
津	Kawai無線OA 津店	(0592)26-0111
四日市	Kawai無線カデンワールド四日市	(0593)54-3366
松坂	Kawai Bax店 6F	(0598)26-0111
大津	西武百貨店関西大津	(0775)25-0111
日本橋	J&Pテクノランド	(06) 634-1211
	J&Pメディアランド	(06) 634-1511
	ニノミヤエレランド	(06) 632-2038
	ニノックスコア日本橋店	(06) 647-2038
難波	ニノミヤパソコンランド難波店	(06) 643-3217
梅田	ニノミヤパソコンランド駅前第4ビル店	(06) 341-2031
	J&P阪急三番街店	(06) 374-3311
高槻	上新電機J&P高槻店	(0726)85-1212
八尾	西武百貨店関西八尾店	(0729)97-0111
京都	J&P京都寺町店	(075)341-3571
和歌山	J&P和歌山店	(0734)28-1441
神戸	星電社三宮本店C-ソフト	(078)391-8171
	ダイエー三宮電器館パレックス	(078)391-7911
	上新電機三宮一番館	(078)211-2111
姫路	上新電機J&P姫路店	(0792)22-1221
	姫路コンピュータソフトコンパックス	(0792)94-8244
岡山	岡山VIVRE 21	(0862)32-8881
倉敷	倉敷ファソランド	(0864)25-4701
広島	松本無線パーツ	(082)243-4451
	ダイイチパソコンCITY	(082)248-4343
高松	Sound Check MOVE	(0878)61-6171
徳島	徳島そごう 7F	(0886)25-4056
高知	デンキのタグチ	(0888)23-0181
松山	ダイイチ松山パソコンCITY	(0899)31-6711
北九州	ベストマイコン小倉パソコン館	(093)551-6281
福岡	ベストマイコン福岡店	(092)781-7131
	寿屋エレデ博多	(092)281-4411
長崎	ベスト電器長崎新地店	(0958)28-1333
熊本	寿屋本荘店	(096)372-5411
	ベストマイコン熊本パソコン館	(096)332-4180
鹿児島	ベスト電器鹿児島パソコン館	(0992)23-2081
大分	ベスト電器大分パソコン館	(0975)32-9396
宮崎	宮崎寿屋百貨店 7F	(0985)27-4111
那覇	ベスト電器那覇店 6F	(0988)62-7588

ブラザー工業株式会社

〒467 名古屋市瑞穂区苗代町2番1号
新事業推進室
TAKERU事務局(052)824-2493

購読方法：定期購読もしくはソフトベンダー武尊（タケル）でお買い求めいただけます。
★定期購読の場合＝定期購読料６ヶ月分6,000円（送料サービス、消費税込）を、
　現金書留または郵便振替で下記の宛先へお送り下さい。
　現金書留の場合：〒171 東京都豊島区要町1―19―3　いさみビル4F　満開製作所
　郵便振替の場合：東京5―362847　満開製作所
●御注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
●新たに購読を開始される方は、「新規」とご明記下さい。
●製品の性格上返品には応じられませんが、お申し出があれば定期購売を解約し残
　金をお返しします。
★武尊でお求めの場合＝１部につき1,200円（消費税込）です。
　●定期購読版と内容が一部異なる場合があります。ご了承下さい。
●お問い合わせ先　TEL(03)3554―9282（月～金　午前11時～午後６時）
（なお、定期購売版のバックナンバーについては定期購読者の方のみご注文を承ります）

船本昇竜（京都府）

プログラマーを目指す者から見た電脳倶楽部の魅力といえば〝はやい・うまい・やすい〟に尽きます。毎月掲載されているプログラムは全部すぐに実行出来るようになっており尚且つ、掲載されている殆どのプログラムのソースが公開されているので自分で新しいプログラムを作る時いろいろと参考になります。それでいて月々千円なんてあまりにもお買い得です。毎月プログラムは非常に高度な物から初心者にでも理解出来そうな物まで数多く掲載されているのでプログラマーを目指す人には必見の〝雑誌〟であるといえます。

赤えんぴつならゴールが見える!!

新発売 赤えんぴつ (JRA版)

最近甘口の予想ばかりとお嘆きの貴兄に、辛口の予想をデータから導く「赤えんぴつ」をそんな貴方にお送りします。
今迄の競馬のコンピュータ用予想プログラムは、オッズを入力して予想するものばかりでした。
この方法はデータ数が少なく入力し易いのですが、オッズは馬券を買った人たちの人気投票的なものですし、貴方の個人的な御意見等も反映出来ず、堅い馬券は時々当たるものの、中穴以上になると7点ぐらい予想をしてもはずれる事が多々あり、回収率も100%を割るものばかりでした。
今回発売した「赤えんぴつ」は当たる馬券を予想するのでは無く、予想紙に載っている馬の過去のデータを入力して、ゴールする時のタイムを予想し上位3頭の馬から3点の組み合わせをはじき出します。
当社で行った過去90回のレースを模擬的に各レース3点で予想した結果では35%の的中率を出し、回収率も130%を上回っています。
過去のデータだけを入力するのでは無く、最新の馬の調子や馬場状態等の主観的なデータも10~100%の数字に置き換えて予想に反映させたり、それらのデータをディスクにセーブする事が出来ますから、レースの前日にデータを入力しておき、レース当日の天候等、直前の情報で各馬のデータを修正して予想を立て直す事も出来ます。
又、コンピュータの苦手な方でも簡単にデータの入力が出来る様にカーソルコントロールキーと実行キーの5つのキーを使うだけで総ての操作が出来ます。
このプログラムはJRA主催の全国10ヶ所（札幌、函館、福島、新潟、中山、東京、中京、京都、阪神、小倉）の各競馬場以外の公営競馬場では使えません。

赤えんぴつ　X68000用　2HD　**20,000円**

便利な超高速通信機能付で、DB. Xよりも使い易く、X1 turboのディスクもアクセス出来る。
SUPER DEVICE MONITOR "T"　X68000用　2HD　**15,000円**
X68000と超高速通信が出来てMS-DOSのディスクや内部増設RAMにもアクセス出来る。
SUPER DEVICE MONITOR "T"　X1 turbo用　2HD/2D　**13,000円**

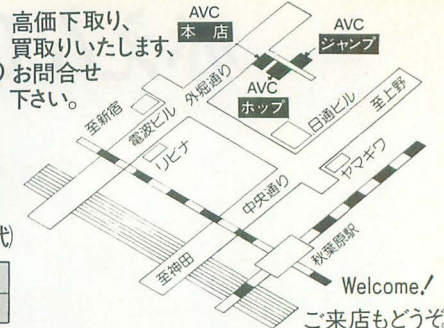
＊MS-DOSはマイクロソフト社の商標です。
＊商品の価格には消費税は含まれていません。

▶お求めは全国の有名マイコンショップでどうぞ。
通信販売をご希望の方は当社へ直接、商品名・機種名・メディア名・住所・氏名・電話番号を明記の上、現金書留にてお申し込みください。(送料無料)

BLUE SKY Co.
株式会社 BLUE SKY
〒411 静岡県三島市加茂16-4 ☎0559-72-6710

AVCフタバ

☎03(3253)7661

〒101 東京都千代田区外神田3-2-3 ☎03-3253-7611(代)

今すぐ もよりの電話から	仙　台 022-264-3704	名古屋 052-452-3271	広　島 082-295-6873
札　幌 011-611-5104	新　潟 0252-75-4175	大　阪 06-311-3931	福　岡 092-481-2494

高価下取り、買取りいたします、お問合せ下さい。

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

X68000 PERSONAL WORKSTATION EXPERTII・EXPERTII HD

待望の新しい仲間登場!!

新年特別セール

EXPERTII・EXPERTII HD
集積度を高めた"マンハッタンシェイプ"3Mの大容量メモリを搭載。本格的なウインドウシステム、SX-WINDOW搭載。

〔写真のモニタは別売です。〕

CZ-603C 標準価格¥338,000
CZ-613C 標準価格¥448,000

AVC特価

X68000 PERSONAL WORKSTATION PROII・PROII HD

PROII・PROII HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。
もちろん、SX-WINDOW搭載。

〔写真のモニタは別売です。〕

CZ-653C 標準価格¥285,000
CZ-663C 標準価格¥395,000

AVC特価

CZ-8PC4

48ドット熱転写プリンター。精密な文字、ハードコピーも可能。

CZ-8PC4……¥ 99,800

AVC特価 ¥64,800

お勧めディスプレイコーナー　組合せは自由、価格はお気軽にご相談下さい。

機種	特徴	機種	特徴	機種	特徴
CZ-604D 標準価格¥94,800 AVC特価	●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカ搭載 ●チルト台同梱	CZ-613D 標準価格¥135,000 AVC特価	●ドットピッチ0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-603D 標準価格¥84,800 AVC特価	●0.31mmドットピッチ ●TVチューナ無し ●3モードオートスキャン ●チルト台同梱
CU-21HD 標準価格¥148,000 AVC特価	●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカ搭載	CZ-605D 標準価格¥115,000 AVC特価	●ドットピッチ0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-602D 標準価格¥99,800 AVC特価	●ドットピッチ0.39mm ●TVチューナー搭載 ●チルト台同梱

X68000 PERSONAL WORKSTATION EXPERT HD

NEW

CZ-604C-TN ……………¥348,000
CZ-606D-TN ……………¥ 79,800
世界標準SCSIインターフェース標準装備。

新年特別セール

1月中にお買上げのお客様にフロッピーディスク20枚と「Vボール」又は「ニュージーランドストーリー」のどちらかをプレゼント!

AVC特価　価格はお電話で

X68000 PERSONAL WORKSTATION SUPER HD

CZ-612C-BK ……………¥466,000
CZ-603D-BK ……………¥84,800
ジョイカード‥¥ 1,400
合計 ¥552,200

新年特別セール

1月中にお買上げのお客様にフロッピーディスク20枚と「Vボール」又は「ニュージーランドストーリー」のどちらかをプレゼント!

超特価 ¥352,000

●但し消費税(3%)は別途請求させていただきます。●分割回数は3回～48回まで自由に選べます。

●頭金なし(手軽な電話クレジット)●製品先取り(お支払いは約1～2ヶ月後から)●低金利クレジット(1回の支払いは2,700円以上で3～48回。ボーナス併用も可)●カレッジクレジット(保証人なし。但し満20歳以上の学生の方)●18歳未満の方(ご両親が代理購入者としてお申し込み下さい)●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい)●完全保証(すべてメーカー保証書付。アフターケア万全)●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時まで受付 日曜・祝日も営業

☎価格は電話で値切って下さい。

株式会社デンキヤ

営業時間AM11:00〜PM7:00 水・木曜定休

(価格は全べて税込みです)

セット超特価 X68000 PERSONAL WORKSTATION PROII・PROII HD

機種	セット¥特価
CZ-653C CZ-604D	¥24,400×12回 ¥13,300×24回
CZ-653C CZ-605D	¥25,700×12回 ¥13,700×24回
CZ-603C CZ-604D	¥27,500×12回 ¥14,600×24回
CZ-603C CZ-605D	¥28,800×12回 ¥15,300×24回

セット超特価 X68000 PERSONAL WORKSTATION EXPERTII・EXPERTII HD

機種	セット¥特価
CZ-604C CZ-606D	¥27,000×12回 ¥14,400×24回
CZ-663C CZ-613D	¥34,000×12回 ¥18,100×24回
CZ-623C CZ-606D	¥35,200×12回 ¥18,800×24回
CZ-623C CZ-613D	¥40,600×12回 ¥21,600×24回

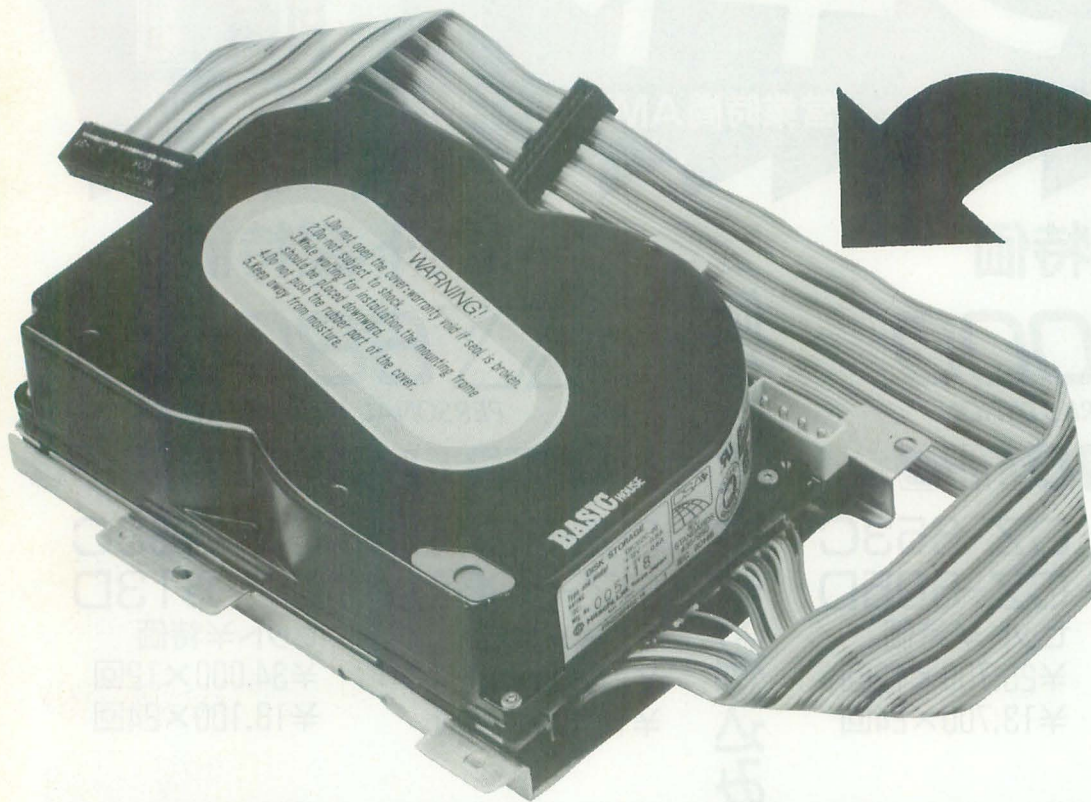
全品メーカー保証 即決クレジットOK

ディスプレイ		プリンタ		周辺機器		ソフト	
CZ-604D	特価	CZ-8PC4	特価	CZ-8NJ1	¥1,400	CZ-213MS	¥15,500
CZ-605D	特価	CZ-8PG1	特価	CZ-8NJ2	¥18,540	CZ-259SS	¥ 5,200
CZ-613D	特価	CZ-8PG2	特価	PIO-6BE1A	¥20,000	CZ-219SS	¥23,100
CU-21HD	特価	JX220	特価	PIO-6BE2	¥39,000	CZ-245LS	¥35,500

24時間テレホンサービス
0482-54-3444

お申し込み
TEL.0482-54-3400
FAX.0482-54-3443
埼玉県川口市西川口4-6-4

お支払い
下記取引銀行口座
までお振込み下さい。
三菱銀行西川口支店
㈱デンキヤ㊢0258081

新X68000誕生！

200M
内蔵

X68000 SUPERにBASIC HOUSEオリジナルSCSI HDを組み込み、
新たなるSUPER HDの誕生です。

X68000/40(40MHD内蔵) ………………… ¥348,000
X68000/100(100MHD内蔵) ………………… ¥398,000
X68000/200(200MHD内蔵) ………………… ¥498,000

※BASIC HOUSEオリジナルなので通信販売のみです。

First Class Technology オリジナル 新製品

X68000用SCSI仕様
200M外付用ハードディスク

「FHD-200」 定価¥298,000

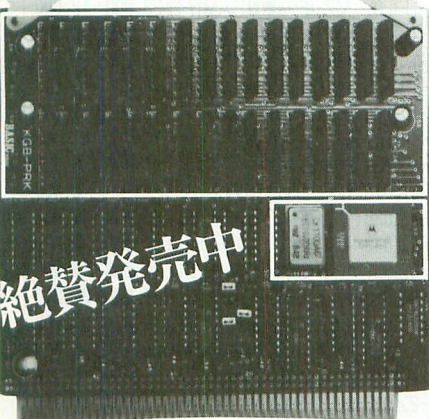
御問合せは、PROSTAFF 登坂まで

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研
本社営業部／マイコンショップ／通販部 宇都宮市竹林町503—1 TEL0286-22-9811 FAX0286-25-397
大田原営業所／マイコンショップ 大田原市美原1—13—4 TEL0287-23-5352 FAX0286-23-536

マイコンショップ BASIC HOUSE お申し込み・お問い合せは ☎0286-22-9811(代)

2枚のボードが1枚になった

KGB-X68PRK

写真はKGB-X68PRK-14です

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2、CZ-6BE4、CZ-6BP1との混在が可能
- 複数枚のKGB-X68PRKの実装が可能
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能
- ジャンパの変更により数値演算プロセッサの1枚目、2枚目、未使用の選択が可能
- 1M、2M、3Mメモリモデルは購入後もメモリ増設が可能
- PRK-10、11、12、13、14にはデバイスドライバ(FLOAT3.X)が付属

※CZ-600C、601C、611C、652C、653C、662C、663Cで御使用の際にはあらかじめ専用の1Mメモリ(CZ-6BE1、A、B等)でメインメモリを2Mバイト以上にしておく必要があります。

製品価格一覧

品名	価格
KGB-X68PRK-00 (メモリ無し/数値演算プロセッサ無し)	¥34,000
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥58,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥74,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥98,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000
KGB-X68PRK-10 (メモリ無し/数値演算プロセッサ付き)	¥76,000
KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥96,000
KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥112,000
KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

PRK質問箱

Q、購入後のメモリ増設はどうやるのでしょう?
A、ご購入後のPRKに対するメモリの増設は半田付け等の技術を要するため原則として当社に送り返していただき増設いたします。自分でメモリ増設をする場合は通信販売のみですが必要な部品の販売も致します。御希望の方はお問い合わせ下さい。

Q、数値演算プロセッサにMC68882を使用することは可能ですか?
A、MC68882では動作しないソフトが存在するため使用できません。

Q、「数値演算プロセッサのみ」や「プロセッサ無しメモリ無し」のPRKがほしいのですが?
A、PRK-10、PRK-00の型番で商品化しております。

※最近PRKをスロットに挿入したが動作しないと言う御質問を良く受けますが、ほとんどの場合は差し込み不足が原因です。X68000のスロットは大変堅く裏蓋が閉まる状態でも差し込み不十分の場合があります。御注意ください。

充実のBASIC HOUSEソフトウェア&ハードウェア

品名	価格
高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥ 68,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800
高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
汎用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
64180CPUボードMach 180(KGB-CPXB) X68000	¥ 98,000
ローコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

品名	価格	品名	価格
BASIC拡張関数パッケージ(B6-6301)	¥9,800	C言語ライブラリ(B6-6305)	¥6,800
ディスクキャッシャー(B6-6304)	¥6,800	Toys & Tools(B6-6307)	¥6,800
BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥14,800		
アイコンエディタ(B6-6303)	¥4,800	CP/M68Kエミュレータ(B6-6302)	¥19,800

おしらせ

DISK CACHER Version UP

皆様に御愛用いただいているディスクキャッシャーが高速化(従来比平均3倍)を行ないVer. UPいたしました。今回のVer. UPはハードディスクキャッシャーのみでHD-DISKCACHE Ver 2.0未満のキャッシャーを御持ちの方がVer. UPの対象となります。
御希望の方は旧バージョンのディスクのラベルと代金¥1,500(送料、税込み)を同封して現金書留で御申し込み下さい。

ビデオボードを外付けに!!
ビデオボードケース(KGB-BVBX)
大好評発売中

SHARPより発売されているCZ-6BV1を外付けにする、ケースです。このケースの使用によりあなたのX68000のスロットが開放されます。

Human68k下のソフトのCRT出力を強制的に15kHZ出力にする(768×512モード除く)おまけユーティリティ付き

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部/マイコンショップ/通販部　宇都宮市竹林町503-1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所/マイコンショップ　大田原市美原1-13-4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

ABIT
アイビット電子株式会社

SHARP

パソコン本体から周辺機器まで品数取り揃え

大特価セール実施中!!

型名	品名	正価	特価
PC-E500BL	ポケコン	28,800	19,800
PC-1600K	ポケコン	69,800	49,800
PC-1360K	ポケコン	36,800	32,800
PC-1360	ポケコン	29,800	19,800
PC-1262	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1280	ポケコン	24,800	19,600
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800
CE-203M	ポケコンRAM32K	32,000	7,000
CE-202M	ポケコンRAM16K	35,000	6,000
CE-201M	ポケコンRAM 8K	18,000	3,000
CE-1600M	ポケコンRAM32K	32,000	16,000
CE-1600F	ポケコンフロッピードライブ	39,800	34,800
CE-1600P	ポケコンプリンター	69,800	59,800
CE-1650F	ポケコンDISK	9,800	8,800
CE-161	ポケコンRAM16K	50,000	3,800
CE-1601M	ポケコンRAM64K	45,000	30,000
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-159	ポケコンRAM 8K	35,000	4,200
CE-140T	ポケコンRS-232Cコンバター	9,800	8,800
CE-140F	ポケコンフロッピディスク	49,800	44,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-120P	ポケコンプリンター	24,800	21,800
CE-124	ポケコンカセットインター	4,500	4,000
Z-VISIONplus	Z80シュミレータ デバッカー	59,800	51,000
UX-1	ホームコピーファクス	78,000	69,800
PA-9500	ハイパー電子手帳	48,000	特価
CZ-300F	X13"マイクロフロッピー	79,800	9,000
CZ-31FS	300F増設フロッピー	59,800	7,000
CZ-82F	CZ-802C増設フロッピー	59,800	6,000
CZ-501H	X1増設用ハードディスクユニット	258,000	60,000
CZ-503F	CZ-830増設ドライブ	49,800	30,000
CZ-520F	2HD/2DDミニフロッピードライブ	118,000	70,000
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIB	1M増設RAMボード	28,000	19,500
CZ-6BE1	1M増設RAMボード	35,000	29,500
CZ-6BE2	2M増設RAMボード	79,800	63,800
CZ-6BE4	4M増設RAMボード	138,000	110,400
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BF1	RS-232C増設ボード	49,800	42,300
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU	RRGBシステムチューナー	33,100	26,500
CZ-6BG1	X6800GPIBボード	59,800	50,000
CZ-6BC1	X6800FAXボード	79,800	67,800
CZ-6PV1	ビデオプリンター	198,000	158,000
CZ-6BV1	ビデオボード	21,000	16,800
CZ-822C	X1G MODEL30	118,000	39,800
CZ-820C	X1G MODEL10	69,800	16,800
CZ-888C	X1TURBO		
CZ-8BGR2	グラフィックボードX1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	X1 漢字ROM	19,800	16,800
CZ-8BM2	232Cマウスボード	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット		
CZ-8BV2	カラーイメージボート	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8TM1	X1ソフト付モデムユニット	29,800	5,000

型名	品名	正価	特価
CZ-8TM2	X1ソフト付モデムユニット	49,800	39,800
CZ-8EB3	拡張i/o box	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	69,000
CZ-8PK7	24ドット80桁漢字プリンター	122,000	59,800
CZ-8PC4GY	24ドット熱転写カラー漢字プリンター	99,800	59,800
CZ-8BS1	X1FM音源ボード	23,800	19,800
CZ-8BK4	X1第2水準ROM	———	5,700
CZ-8NJ2	インテリジェントコントローラー	23,800	18,500
CZ-8NS1	カラーイメージスキャナー	188,000	149,000
AN-S100	アンプ付スピーカー	36,600	29,500
HXD040	アイティム40Mハードディスク(ITM)	118,000	95,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	75,800
CU-14FD	カラーディスプレーアナログ0.31	74,800	49,800
MZ-1D10	12"モノクロディスプレー	41,800	25,000
MZ-1D17	15"CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000 プリンターI/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレルI/F	34,800	28,000
MZ-1E45	MZ6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレル I/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス 300BT	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500 QD用インターフェイス	9,800	3,000
MZ-1M01	MZ-2000/2200 16ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	72,000
MZ-80P4B	136桁ドットプリンター	———	48,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P28	ドットプリンター漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター136桁	168,000	134,400
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500 増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-28611M増設RAM	45,000	15,000
MZ-1R35	MZ-28611M増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R27A	MZ-2500VRAM	13,000	10,000
MZ-1R26A	MZ-2500 増設RAM	15,000	12,800
MZ-1R21	漢字ROM	38,000	13,000
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000
MZ-1R32	MZ-6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000

型名	品名	正価	特価
MZ-1R29	MZ-1P22 増設RAM17-22	32,000	12,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1U09	MZ-2500 拡張ボード	9,000	7,200
MZ-1V01	パソコン FAX	278,000	85,000
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z016	MZ-5500 附属	———	5,000
MZ-2Z028	MZ-6500 MSDOS GWBASIC	60,000	35,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属	———	5,000
MZ-2Z013	MZ-5500 MSDOS	25,000	20,000
MZ-4Z001	MZ-5500 IBM変換	30,000	8,000
MZ-5521	本体	388,000	55,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト	———	3,500
MZ-6F03	ブランク QD DISK	450	400
MZ-6P18	MZ-1P18、28カットシートフィーダー	60,000	35,000
MZ-6P11	MZ-1P10 カットシード	95,000	35,000
MZ-6P29	MZ-1P29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,000	2,700
MZ-6Z22	MZ-6500(50)CP/M86BASIC-3	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8B104	MZ200/2200 GP IBインターフェイス	45,000	18,000
MZ-8BG	MZ-80BGRAM1	39,000	10,000
MZ-8BC01	MZ200/2200 GP IBケーブル	18,000	8,000
UE-1U01	X286L スロットBOX	5,000	4,000
UE-1R02	4M RAMボード	300,000	240,000
UE-1R06	辞書ROMボート	32,800	25,600
UE-1R01	2M RAMボード	160,000	128,000
UE-1R05	拡張グラフィックボード	92,000	55,000
UE-1R03	1M RAMボード	100,000	80,000
UE-1R04	2M RAMボード	180,000	144,000
UE-1P03	80桁漢字プリンタ	———	特価
UE-1P04	136桁漢字プリンタ	———	特価
UE-1P05	136桁漢字水平プリンタ	———	特価
UE-1P02	高速136桁漢字プリンタ	550,000	440,000
UE-1P01	136桁漢字プリンタ	268,000	214,400
UE-1E04	S-RNインターフェイスカード	70,000	56,000
UE-1E02	AX286L ICカードI	45,000	36,000
UE-1E03	5"FDインターフェイスカード	28,000	22,400
UE-1D03	15インチカラーディスプレイ	123,000	98,400
UE-1D02	14インチカラーディスプレイ	158,000	126,400
IO-735X	カラープリンター	248,000	195,000
BF-68PRO	フィルター	19,800	16,800
X6800キーボード延長ケーブル(1.5m)		2,500	2,000
ディスプレーケーブルアナログ15P(3m)		5,000	4,000
ディスプレーケーブルアナログ15P(1.5m)		4,300	3,500

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

FM TOWNS/FM NOTE/東芝ダイナブック、周辺機器も取扱っております。

X68000全機種取り揃え大特価セール

シャープパソコンフェア '91 1/19・20開催

'91年2月末迄

ワープロ、パソコンお買い上げの方は、ワープロ、パソコン教室が御利用になれます。

SHARP X68000シリーズ対応 ハードディスク

〈ITEM〉

HXD 040 23mS X68000
定価￥118,000⇒特価￥95,000

HXD 042 X68000 増設用
定価￥128,000⇒特価￥102,500

HXD 140 X68000 内蔵用
定価￥98,000⇒特価￥79,800
（602・603はHXD-140に内蔵）

PIXELA
MacIIフルカラー・イメージクリエータ〈ピクセカラー735〉
定価 ￥128,000
新発売！入荷

SHARP
光磁気ディスクドライブ JY-7000
新発売！入荷

SHARP X68000 特価表示は、オープン記念特別価格にてご提供いたします。

特価表示はTELにてご確認下さい。

CZ-602C（本体） プラス・40Mハードディスク付

CZ-603D	￥315,000
CZ-602D	￥350,000
CZ-612D	￥365,000
CZ-613D	￥375,000

CZ-602C（本体） プラス（ディスプレイ）組合せ

CZ-606D	￥270,000
CZ-613DGY	￥310,000
CZ-605DGY	￥300,000
CZ-611DGY	￥285,000

CZ-603C（本体） プラス・40Mハードディスク付

CZ-603D	￥365,000
CZ-602D	￥380,000
CZ-612D	￥395,000
CZ-613D	￥400,000

CZ-603C（本体） プラス（ディスプレイ）組合せ

CZ-602DBK	特 価
CZ-606D	特 価
CZ-611DGY	￥305,000
CZ-613D	特 価

CZ-604C（本体） プラス（ディスプレイ）組合せ

CZ-604D	特 価
CZ-611DGY	￥310,000
CZ-606D	特 価
CZ-605D	特 価

CZ-623CTN（本体） プラス（ディスプレイ）組合せ

CZ-611DGY	￥445,000
CZ-612DGY	￥460,000
CZ-613DTN	特 価
CZ-21HD	特 価

CZ-652C（本体） プラス（ディスプレイ）組合せ

CZ-602DBK	￥275,000
CZ-606D	￥260,000
CZ-612DGY	￥290,000
CZ-605D	￥290,000

CZ-653C（本体） プラス（ディスプレイ）組合せ

CZ-602DBK	特 価
CZ-606D	特 価
CZ-612DGY	￥290,000
CZ-605D	特 価

パソコンゲームソフト（X1、X1t対応）

		特 価
トンネルズ&トロールズ	X1/X1t	￥8,330
ロードウォー2000	X1t	￥8,330
イースII	X1t	￥6,630
ソーサリアン	X1t	￥8,330
ソーサリアン(ユーティリティー)	X1t	￥3,230
ソーサリアン No.1	X1t	￥3,230
ソーサリアン No.2	X1t	￥3,230
ソーサリアン No.3	X1t	￥3,230
三国志II	X1t	￥12,580
ラスト・ハルマゲドン	X1t/Z	￥6,630
ランペルール	X1t	￥8,300
ザナドゥ	X1/X1t	￥6,630
水滸伝	X1t	￥8,330
大航海時代	X1t	￥8,330
アークス	X1t	￥8,330
信長の野望（群雄伝）	X1t	￥8,330
エグザイル	X1t	￥7,480
上海	X1/X1t	￥5,525
マスターオブモンスターズ	X1t	￥6,800
ウイザードリー	X1/t	￥8,330
パワフルまあーじゃん2	X1t	￥6,630
サイオブレイド	X1t	￥7,480
ザナドゥ シナリオII	X1t	￥4,930
琥珀色の遺言	X1t	￥8,330
倉庫番	X1/t	￥5,780
INKPOT	X1/t	￥15,300
マシン言ゲーム	X1/t	￥3,660
構造化BASICのすすめ	X1/t	￥3,660
マクロアセンブラー MACRO-80	X1/t	￥17,500
信長の野望 全国版	X1/t	￥8,330
ファンタジーIII	X1/t	￥8,330
マイト・アンド・マヂック2	X1	￥8,330
ビジレス	X1t	￥40,800
デバイスモニター	X1/t	￥4,900
JETターボターミナル	X1t	￥8,330
金庫番	X1/t	￥9,000
麻雀悟空	X1	￥5,780
ダ・ビンチ(スーパーグラフィック2)	X1/X1t	￥5,780
CZ-161LF C	X1/t	￥11,700
CZ-115LF FORTRAN	X1/t	￥11,700

アイビット推奨ディスプレイ

シャープ
CZ-611DGY
ドットピッチ0.31
チルト台付
特価￥79,800

シャープ
CZ-880D
特価￥59,800

シャープ
CZ-602D-BK
（15型アナログTV/3モードオートスキャン）
特価￥75,000

シャープ
21G-SF1
スーパーファミコン内蔵テレビ
特価￥125,000

※シャープ周辺機器（拡張、プリンター他）も常時取り扱っております。

富士通 FM R50 FM NOTE
➡特価￥190,000

SHARP AX286N-H2 All In Note
➡特価￥358,000
外付けドライブユニット3.5インチプレゼント！

NEC PC-98HA NOTE
➡特価￥155,000

TOSHIBA J-3100S S001 DynaBook
定価￥198,000
➡特価￥99,800
キャリングケースサービス

富士通FM TOWNSお買得セット

FM TOWNS モデル2基本セット

FM TOWNS-2	￥398,000
FMT-DP533	￥ 69,800
FMT-KB101	￥ 20,000
B276A010	￥ 20,000
定価合計	￥507,800

特価￥228,000

FM TOWNS 1S拡張2セット

FM TOWNS-1S	￥338,000
HM-01T	￥ 32,800
FMT-DP533	￥ 69,800
FMT-KB101	￥ 20,000
B276A010	￥ 20,000
定価合計	￥480,600

特価￥208,000

FM TOWNS 1拡張2セット

FM TOWNS-1	￥338,000
HM-01T	￥ 32,800
FMT-DP533	￥ 69,800
FMT-KB101	￥ 20,000
B276A010	￥ 20,000
FMT-FD301	￥ 28,000
定価合計	￥508,600

特価￥218,000

FM TOWNS 2F基本セット

FM TOWNS-2F	￥378,000
FMT-KB101	￥ 20,000
FMT-DP533	￥ 69,800
B276A010	￥ 20,000
定価合計	￥487,800

特価￥278,000

FM TOWNS 2F基本セット

FM TOWNS 2F	￥378,000
FMT-DP533	￥ 69,800
FMT-KB101	￥ 20,000
B276A010	￥ 20,000
定価合計	￥487,800

特価￥278,000

FM TOWNS 20F基本セット

FM TOWNS 20F	￥323,000
FMT-DP533	￥ 69,800
FMT-KB101	￥ 20,000
B276A010	￥ 20,000
定価合計	￥432,800

超特価

〈TOWNSお買い上げの方〉パソコン教室が御利用できます。初・中・上級者 無料にて実施中！

〈全商品新品完全保証付〉■シャープポケコン全商品販売中。カタログ、特価表ご請求ください（〒72）

0426-45-3002（駅前店）-3001（本 店）-3003（教 室）

FAX.0426-44-6002

●営業時間/10:00〜19:00●電話受付/20:00迄可●定休日/水曜日

SHARP SUPER XEX SHOP

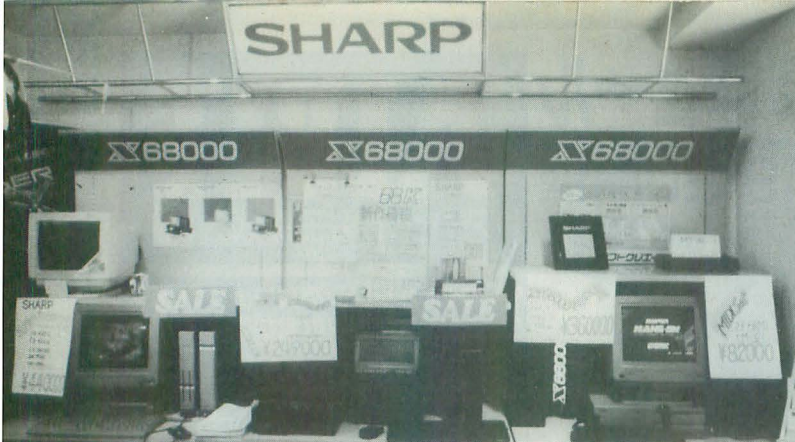
アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。

全国通信販売 北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 （普）1752505

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00〜PM7:00

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

新春大感謝セール!

X68000 NEW PRO II

- ●CZ-653C(本体)……………¥285,000
- ●CZ-60 D(カラーディスプレイ)……………¥ 84,800
- ●お好きなゲームソフト1本……………¥ 7,800
- ■定価合計……………¥377,600

クリエイト特価

均等払い	¥ 7,680×48回	¥ 9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 EXPERT セット

5 台 限 定 !

- ●CZ-602C-GY(本体)……………¥356,000
- ●CZ-603D-GY(カラーディスプレイ)……………¥ 84,800
- ■定価合計……………¥440,800▶大特価¥279,000

大特価¥279,000

均等払い	¥12,850×24回	¥ 8,870×36回	¥ 6,920×48回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- ●CZ-603C(本体)……………¥338,000
- ●CZ-613D(カラーディスプレイテレビ)……………¥135,000
- ●CZ-8NJ2……………¥ 23,800
- ●お好きなゲームソフト1本……………¥ 9,800
- ■定価合計……………¥506,600

クリエイト特価

均等払い	¥ 9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 SUPER HD

- ●CZ-623C-TN(本体・キーボード・マウス)……………¥498,000
- ●CZ-613D-TN(カラーディスプレイ)……………¥135,000
- ●CZ-6BP1……………¥ 79,800
- ■定価合計……………¥712,800

クリエイト特価

均等払い	¥ 7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000× 8回	¥50,000× 6回	¥80,000× 4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ〜/

- ●CZ-603C……………¥338,000
- ●CZ-605D……………¥115,000
- ●MU1.B(MIDIボード&ソフト)……………¥ 39,800
- ●CM32L……………¥ 69,000
- ●グラナダ……………¥ 8,800
- ●JOYカード……………¥ 1,800
- ■定価合計……………¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET/

- ●PRO II CZ653C……………¥285,000
- ●0.31CRT CZ603D……………¥ 84,800
- ●グラナダ……………¥ 8,800
- ●Y'S……………¥ 8,700
- ●ポピュラス……………¥ 9,800
- ●スーパーハングオン……………¥ 8,800
- ●エージャックス……………¥ 8,800
- ●サーク……………¥ 8,800
- ●アールタイプ……………¥ 7,800
- ●アナログJOYSTIC XE-1AP……………¥ 13,800
- ■定価合計……………¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO	MIDI版	¥ 28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥ 28,800
CZ-8NM3	マウス・トラックボール	¥ 9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
BF-68PRO	高性能CRTフィルター	¥ 19,800	PRO-68K	サイバーノート	¥ 19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	PRO-68K	ステーショナリー	¥ 14,800
CZ-8NT1	トラックボール	¥ 13,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6BM1	MIDIボード	¥ 26,800	Human 68K Ver2.0	開発ツールセット	¥ 9,800
CZ-8NJ2	アナログスティック	¥ 23,800	PIO-6BE1-A	内蔵1MRAM	¥ 25,000
CZ-6TU	パソコンチューナ	¥ 33,100	PIO-6BE2-2M	2MRAM	¥ 50,000
SX-68M	MIDI I/F	¥ 19,800	PIO-6BE4-4M	4MRAM	¥ 88,000
XE-1AP	アナログジョイパッド	¥ 13,800	MU1-B	MIDI I/F + ソフト	¥ 39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。 ●超特価販売中!

オール15%〜20%OFF

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

●渋谷店

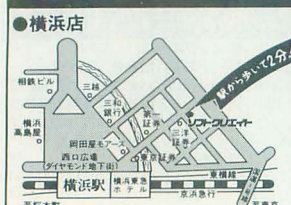
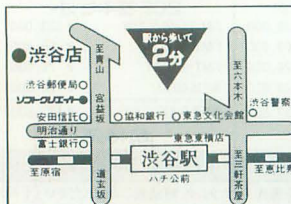
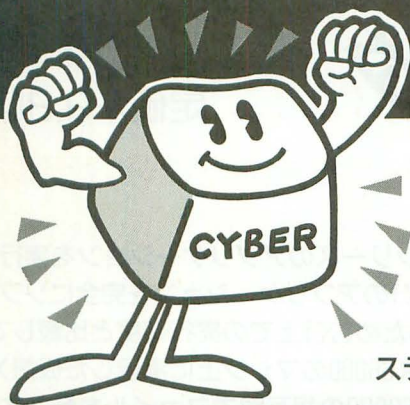
●横浜店

総合お問合せ先☎03-3486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-3486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:太陽神戸三井銀行 渋谷宮益坂支店(普)No.5000340

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

このキーボードは一味違う!!

あなたの X68000 のキーボードをチューンナップします。

ステージI…合計94個のキースイッチをクリック感抜群の物と交換!!
ステージII…ステージI＋キーボードの101箇所に(誤)入力防止処理を施します。

ご注意

● LED付のキー7個、BREAK・COPYキー、F1〜F10キー は構造上変更出来ません。
その他の入力に必要なキーを変更します。
● X68K PRO・PROIIには対応していません。

メニュー

ステージI… ¥19,800
ステージII… ¥29,800

● 当社からの発送代金は全てサービスです。
● 消費税は、いただいておりません。

通信販売のみ

ご注文は、住所・氏名・年齢・TEL・御支払方法そして、ステージIかステージIIかを選んで、TEL・FAX・はがき等でお申し込み下さい。

御支払方法 1. 現金書留・郵便為替
2. 郵便振替 横浜4-31963
3. 銀行振込 協和銀行 狛江支店 当座 009867

入金確認しだい梱包用の箱をお送りしますので、あなたのキーボードを入れて御返送下さい。
当社に着きしだいすぐに作業にかかり、約一週間でお手元にお届け致します。

CYBER Corp.
株式会社 サイバー 〒227 横浜市緑区鴨志田町801-32
お問い合わせは、お気軽に TEL. 045(962)1447 FAX. 045(962)1457

第2種 情報処理技術者講座

▶総合コース―8ヵ月
▶受験コース―6ヵ月 (両コース共、6ヵ月延長可能。)

特別優待受講制度

◆学生の方には、特別学割受講制度があります。
◆企業における集団受講(3名以上)の場合にも割引制度があります。

本講座の五大特色

1 入門コース併設で、初心者の方でもやさしくマスターできます。
2 試験合格にマトを絞った実戦的オリジナルテキストで、アセンブラ言語CASLにも対応。
3 プログラミング言語は、実務・受験に有利なフォートランかコボルのどちらかを選択できます。
4 駿台電子ベテラン講師陣による受験(13回)、総合(15回)におよぶ個人別添削指導で、特に合格の決め手になる「プログラミング」を徹底指導。
5 企業研修の一環として受講される場合、労働省「生涯能力開発給付金制度」の適用が受けられます。

★ハガキか電話で下記まで。

学校法人 駿河台学園 〒101 東京都千代田区神田駿河台2-1-20 お茶の水ユニオンビル5F
駿台電子情報専門学校 通信教育部 ohX①係 TEL 03(3295)5042 FAX 03(3293)3739

ACCESS

X1エミュレータ

好評発売中

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して平均3～5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

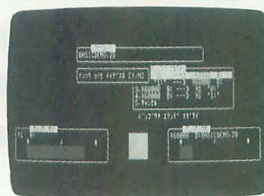
X1ディスク⟷X68000 Human68k(5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC:CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

*タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
*一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

*この商品価格には消費税は含まれておりません。
*CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。
*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

資料請求券 oh!X 2月号

有限会社アクセス 〒101 東京都千代田区神田神保町1-6 神保町協和ビル7 ☎03(3233)0200(代) FAX.03(3291)70

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

SIG探訪PART7

こんな人、ぜひおいで〜

心理学を志す人、心理学に興味のある人、パソコン通信ならではの「心と心の会話」を楽しみたい人、子育てまっ最中のパパ&ママと、その予備軍(!?)、etc.

サイコロジスト

(ジャンプコード:PSY101)

年に3〜4回の大きなオフラインミーティングには30名近い老若男女が集まります。これには家族で参加する人も多く、幼児がまわりを走りまわっている中での宴となります。又、他のSIGと姉妹SIGの提携を結び、合同でオフラインミーティングを開き、SIG間交流も盛んに行っています。

◀大規模ミート以外にも、機会あるごとに集まっています。
ときには、学会の場でアカデミックなミニミートにも……
(左からキョロ・髭・デンタ・らっぱ・俊〈Nifty FPSY「こころとからだボード」OP〉・ミミミ)

人と人、心と心のつながりを大切に!

サイコロジストとは心理学者のこと。メンバーには心理学を専門とする人が多いけれど、ボードが専門用語一色というワケではありません。なによりも心(MIND)のつながりを大切にする中年(!?)の集まりです。そして、まだまだメディアとして市民権を得ていないパソコン通信を、新しい文明の利器として使いこなしたいと願うメンバーが多いのも特徴。ボードを使って専門書出版の打ち合わせ・編集作業をすすめ、「まばたきの心理学(北大路書房)」の出版にこぎつけました。「子供の健康とコンピュータ」という全国規模のアンケート調査の連絡にもボードを使用しています。全国各地の仲間と1つの仕事を行うのに有利なパソ通。ワーキングボードとしての意義に注目し、実証していこうと努力しています。

その他 楽しいメニューがまだまだいっぱい!

★J&Pならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING**。
★J&Pだから強い!!パソコン情報をはじめとする役に立つ**DATA BASE**。
★みんなでおしゃべり**オンライントーク**(CHAT機能)。
★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
★ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03)3496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4−39−1	☎(0425)36-4141
本厚木店	厚木市中町3−4−3	☎(0462)25-1548
富山店	富山市桜町2−1−10	☎(0764)32-3133
金沢店	金沢市入江2−63	☎(0762)91-1130
寺地店	金沢市寺地2−3	☎(0762)47-2524
大須店	名古屋市中区大須4丁目2-48	☎(052)262-1141
新テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06)634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06)634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06)634-3111
U.S.LAND	大阪市浪速区日本橋4丁目9番15号	☎(06)634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06)348-1881
梅田店	大阪市北区小松原町1−10	☎(06)362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06)834-4141
摂津富田店	高槻市大畑町24−10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4−20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451−3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5−11	☎(0798)71-1171
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478−1	☎(0742)27-1111
郡山インター店	大和郡山市横田693−1	☎(07435)9-2221
熊本店	熊本市手取本町4−12	☎(096)359-7800

ひらかれた知性。

X68000
SX-WINDOW

見はてぬ夢の象徴。
次代のインテリジェンス、"SX-WINDOW"搭載。
パーソナルワークステーションのステイタス――X68000。

PRO II
SUPER HD
EXPERT II

PERSONAL WORKSTATION

SUPER・EXPERT・PRO

NEW
X68000SUPER 新発売

ザ・ワークステーション。80Mバイト(SCSI仕様)ハードディスク(HDタイプ)、
SCSIインターフェイスを標準装備。
SUPER 本体+キーボード+マウス・トラックボール
CZ-604C-TN(チタンブラック) 標準価格348,000円(税別)
HDタイプ CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

アートの系譜。**EXPERT II** 本体+キーボード+マウス・トラックボール
CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンダード。**PRO II** 本体+キーボード+マウス
CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)/HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

●ディスプレイは、別売です。

資料請求券 X68000 Oh!X 2係

T4910217902560 雑誌 02179-2